

TWO DIMENSIONAL CHANGE DETECTION METHODS FOR SATELLITE IMAGES

by  
Murat İlsever

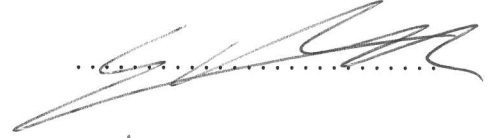
Submitted to the Institute of Graduate Studies in  
Science and Engineering in partial fulfillment of  
the requirements for the degree of  
Master of Science  
in  
Computer Engineering

Yeditepe University  
2012

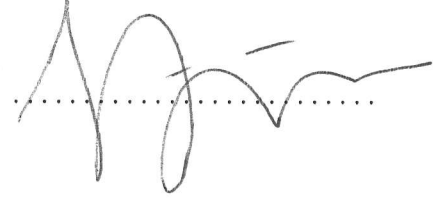
## TWO DIMENSIONAL CHANGE DETECTION METHODS FOR SATELLITE IMAGES

APPROVED BY:

Assoc. Prof. Dr. Cem Ünsalan  
(Thesis Supervisor)



Assoc. Prof. Dr. Sezer Gören Uğurdağ



Assist. Prof. Dr. Dionysis Goularas



DATE OF APPROVAL: .../.../....

## **ACKNOWLEDGEMENTS**

I would like to thank to my advisor, Cem Ünsalan, for his unlimited support. I would also thank to TÜBİTAK for financially supporting my thesis study under project no 110E302.

## **ABSTRACT**

### **TWO DIMENSIONAL CHANGE DETECTION METHODS FOR SATELLITE IMAGES**

Change detection using remotely sensed images has many applications such as urban monitoring, land-cover change analysis, and disaster management. Therefore, we investigated two dimensional change detection methods in this thesis study. We grouped the existing methods in the literature into four categories as: pixel based, transformation based, texture analysis based, and structure based. Besides testing the existing ones, we introduce four new change detection methods as: fuzzy logic based, shadow detection based, local feature based, and bipartite graph matching based. The last two novel methods are the basis for structural analysis of change detection. We compare three thresholding algorithms and measure their effects on the performance of change detection methods. In testing the existing and newly introduced change detection methods, we use a total of 35 panchromatic and multi-spectral Ikonos image sets. We provide quantitative test results as well as their interpretation.

## ÖZET

# UYDU İMGELERİNDE İKİ BOYUTLU DEĞİŞİM BULMA YÖNTEMLERİ

Uzaktan algılanan görüntülerle deęişim bulunması kentsel gelişimin izlenmesine yardımcı olmakta, deprem, sel gibi doğal felaketlerde hasar tespiti yapılmasını sağlamaktadır. Bu tez çalışmasında iki boyutlu deęişim bulma yöntemleri ele alınmıştır. Çalışmamızda yer alan yöntemleri dört kategori altında topladık: piksel tabanlı yöntemler, dönüşüm tabanlı yöntemler, doku analizi tabanlı yöntemler ve yapı tabanlı yöntemler. Bu çalışmada literatürde bilinen yöntemleri test etmenin yanı sıra dört yeni deęişim bulma yöntemi öneriyoruz: bulanık mantık tabanlı yöntem, gölge karşılaştırması tabanlı yöntem, yerel öznitelik tabanlı yöntem ve graf eşleştirmesi tabanlı yöntem. Önerdiğimiz son iki yöntem yapı tabanlı deęişim bulmanın temelini oluşturmaktadır. Deęişim bulmada kullanılan eşikleme algoritmalarından üçünü karşılaştırdık ve deęişim bulma yöntemlerinin performansına etkilerini ölçtük. Yöntemlerin test edilmesinde toplam 35 pankromatik ve multispektral Ikonos görüntü çifti kullandık. Çalışmanın sonunda test sonuçlarını ve yorumlarını verdik.

## TABLE OF CONTENTS

ACKNOWLEDGEMENTS .....	iii
ABSTRACT .....	iv
ÖZET .....	v
LIST OF FIGURES .....	viii
LIST OF TABLES .....	xii
LIST OF SYMBOLS/ABBREVIATIONS .....	xiii
1. INTRODUCTION .....	1
1.1. Literature Review on Change Detection in Satellite Images .....	1
1.2. Layout of the Thesis .....	5
2. PIXEL BASED METHODS .....	7
2.1. Image Differencing .....	7
2.2. Image Ratioing .....	13
2.3. Image Regression.....	14
2.4. Change Vector Analysis .....	16
2.5. Pixelwise Fuzzy XOR Operator .....	18
3. TRANSFORMATION BASED METHODS .....	23
3.1. Principal Component Analysis (PCA).....	23
3.2. Kauth-Thomas Transformation.....	25
3.3. Vegetation Index Differencing.....	27
3.4. Time Dependent Vegetation Index.....	30
3.5. Color Invariants .....	34
4. TEXTURE ANALYSIS BASED METHODS.....	37
4.1. Gray Level Co-occurrence Matrix .....	37
4.2. Entropy .....	41
5. STRUCTURE BASED METHODS .....	44
5.1. Edge Detection .....	44
5.2. Gradient Magnitude based Support Regions (GMSR).....	45
5.3. Matched Filtering.....	45
5.4. Mean Shift Segmentation .....	46

5.5. Local Features .....	48
5.6. Bipartite Graph Matching .....	49
5.7. Shadow Information .....	55
6. EXPERIMENTS .....	57
6.1. The Data Set .....	57
6.2. Performance Tests .....	57
6.2.1. Pixel Based Change Detection Methods .....	63
6.2.2. Transformation Based Change Detection Methods .....	64
6.2.3. Texture Based Change Detection Methods .....	66
6.2.4. Comparison of Thresholding Algorithms.....	66
6.2.5. Structure Based Change Detection Methods.....	70
7. CONCLUSIONS.....	73
REFERENCES .....	74

## LIST OF FIGURES

Figure 2.1.	Distribution of a difference function. Significant changes are expected at the tails of the distribution.....	8
Figure 2.2.	Images taken at two different times from a developing region of Adana...	11
Figure 2.3.	Image differencing applied to the Adana image set. ....	11
Figure 2.4.	Image ratio applied to the Adana image set.....	13
Figure 2.5.	Image difference after regression applied to the Adana image set. ....	15
Figure 2.6.	Unchanged and changed pixel vectors in a 2-D spectral space .....	16
Figure 2.7.	Change vector space divided into four subsets.....	17
Figure 2.8.	CVA (in terms of magnitude value) applied to the Adana image set. ....	18
Figure 2.9.	Fuzzy XOR applied to the Adana image set.....	22
Figure 3.1.	Difference of principal components for the Adana image set. ....	26
Figure 3.2.	Difference of KTT bands for the Adana image set. ....	28
Figure 3.3.	Difference of RVI, NDVI, TVI and SAVI for the Adana image set.....	31
Figure 3.4.	$\psi'_t, \psi''_t, \theta'_t$ and $\theta''_t$ applied to the Adana image set. ....	33
Figure 3.5.	Difference of $c_1, c_2, c_3$ for the Adana image set .....	36



Figure 3.6.	Thresholded difference of $c_2$ for the Adana image set.....	36
Figure 4.1.	Pixels in red circles are adjacent and distance between them is two pixels	37
Figure 4.2.	Four adjacent pixels of a pixel from four directions.....	38
Figure 4.3.	Image differencing applied on GLCM features.....	42
Figure 4.4.	Entropy images calculated for test images .....	42
Figure 4.5.	Change detection by entropy texture feature. ....	43
Figure 5.1.	Edge maps obtained from the Adana image set .....	44
Figure 5.2.	Change detection results from the edge map matching of the Adana image set.....	45
Figure 5.3.	GMSR obtained from the Adana image set.....	45
Figure 5.4.	Change detection results from the GMSR matching of the Adana image set .....	46
Figure 5.5.	GMSR obtained from the Adana image set.....	46
Figure 5.6.	Change detection results from the GMSR matching of the Adana image set .....	47
Figure 5.7.	Segments obtained from the Adana image set .....	47
Figure 5.8.	Change detection results from the segments of the Adana image set .....	48

Figure 5.9.	Change detection using SIFT features.....	50
Figure 5.10.	Change detection using FAST features.....	51
Figure 5.11.	Change detection using the combination of FAST and SIFT features. ....	52
Figure 5.12.	Local features extracted by FAST which are used in graph formation.....	54
Figure 5.13.	Graph matching based comparison results. ....	54
Figure 5.14.	$\psi_b$ applied to test images .....	56
Figure 5.15.	$\psi_b$ results are threshold by Kapur's algorithm .....	56
Figure 5.16.	Shadow differences plotted on the second image .....	56
Figure 6.1.	Three sample image pairs acquired from Adana and Ankara.....	58
Figure 6.2.	Ground truth images for three test image pairs: a. <i>AdanaP<sub>2</sub></i> ; b. <i>AnkaraP<sub>3</sub></i> ; c. <i>AnkaraP<sub>10</sub></i> .....	59
Figure 6.3.	Pixel correspondence between the ground truth set and the result set .....	62
Figure 6.4.	Thresholding algorithms are compared in terms of pixel based methods..	68
Figure 6.5.	Thresholding algorithms are compared in terms of PCA and KTT .....	68
Figure 6.6.	Thresholding algorithms are compared in terms of vegetation indices, TDVIs and color invariant .....	69

Figure 6.7. Thresholding algorithms are compared in terms of texture based change  
detection methods ..... 69

## LIST OF TABLES

Table 6.1.	Ground truth pixel information for the panchromatic test image set. ....	60
Table 6.2.	Ground truth pixel information for the multi-spectral test image set. ....	61
Table 6.3.	Ground truth total pixel information for panchromatic images and multi-spectral images. ....	61
Table 6.4.	Performance measures for pixel based methods using Otsu's method. ....	64
Table 6.5.	Performance measures for transformation based methods using Otsu's method. ....	65
Table 6.6.	Performance measures for texture based methods using Otsu's method. ....	66
Table 6.7.	Effect of the percentile value on the performance measures. ....	67
Table 6.8.	Performance measures for the structure based methods. ....	71

## LIST OF SYMBOLS/ABBREVIATIONS

$b$	Blue band
$br$	Brightness band of KTT
$C_x$	Covariance matrix
$E$	Entropy
$e_2$	Squared error
$g$	Green band
$gr$	Greenness band of KTT
$GT$	Ground truth set
$H_I$	Histogram of image $I$
$H(I)$	Shannon entropy of image $I$
$\hat{I}$	Regressed image
$\tilde{I}_i$	Normalized form of $I_i$
$I_d$	Difference image
$I_i$	Image from $t_i$
$I_r$	Ratio image
$n$	Near infrared band
$N$	Total number of pixels in image
$N_g$	Number of gray levels
$n_i$	Number of pixels at gray level $i$
$ns$	None-such band of KTT
$P$	Percentile
$P(.)$	Probability function
$p_i$	Relative frequency of gray level $i$
$r$	Red band
$RE$	Result set
$T$	Binary image
$t_i$	Time $i$
$Y$	Principal component matrix

$ye$	Yellowness band of KTT
$\theta_t$	Time dependent normalized vegetation index
$\mu_{H_I}$	Mean of the histogram of image $I$
$\mu_i$	Mean of $I_i$
$\mu_I(\cdot)$	Fuzzy membership function of image $I$
$\sigma_b^2$	Between-class variance
$\sigma_{H_I}$	Standard deviation of the histogram of image $I$
$\sigma_i$	Standard deviation of image $I_i$
$\tau$	Threshold value
$\psi_t$	Time dependent ratio vegetation index
$\omega(k)$	class probability up to gray level $k$
BF	Branching Factor
CVA	Change Vector Analysis
DP	Detection Percentage
FAST	Features from Accelerated Segments Test
GLCM	Gray Level Co-occurrence Matrix
GMSR	Gradient Magnitude based Support Regions
KTT	Kauth-Thomas Transformation
LoG	Laplacian of Gaussian
MSAVI	Modified Soil Adjusted Vegetation Index
MSS	Multi-spectral Scanner
NDVI	Normalized Difference Vegetation Index
PCA	Principal Component Analysis
PCC	Percent Correct Classification
RVI	Ratio Vegetation Index
SAVI	Soil Adjusted Vegetation Index
SIFT	Scale Invariant Feature Transform
TDVI	Time-dependent Vegetation Index

# **1. INTRODUCTION**

Change detection is the process of identifying differences in a region by comparing images taken at different times. It finds applications in several fields such as video surveillance, medical imaging, and remote sensing (using satellite imagery). Especially for remote sensing applications, manually labeling and inspecting changes is a cumbersome task. Also, manual inspection is prone to errors and is highly subjective depending on the expertise of the inspector. On the other hand, several applications of change detection using satellite images are land use and cover change, forest or vegetation change, change after forest fire, wetland change, urban change and flood monitoring. Therefore, in this thesis study we focus on automated change detection methods for remote sensing applications using satellite images.

## **1.1. LITERATURE REVIEW ON CHANGE DETECTION IN SATELLITE IMAGES**

We start by giving a survey of change detection review articles in the literature. In the following sections, we explore the existing methods in detail. In this section, we benefit from these survey articles by their comparative results and summary of the advantage and disadvantage of each method in the literature.

In satellite image based change detection applications, the resolution is one of the most important factors. While the first earth observation satellites (such as LANDSAT) were equipped with 30-100 meter resolution sensors; modern sensors can capture images up to 0.5 meter resolution. This also led to the evolution of change detection methods for satellite images. Early change detection methods were generally pixel based. As the detail in the image increased, more sophisticated approaches emerged (such as feature based methods) for change detection. Therefore, initial survey papers only focused on pixel based methods.

Singh [1] summarized several change detection methods such as image differencing, image regression, image ratioing, vegetation index differencing, Principal Component Analysis

(PCA), post-classification comparison, and change vector analysis in terms of land cover change. Singh indicated the relationship between the land cover change and the intensity values of the satellite images as: “The basic premise in using remote sensing data for change detection is that changes in land cover must result in changes in radiance values and changes in radiance due to land cover change must be large with respect to radiance changes caused by other factors”. These other factors are counted as differences in atmospheric conditions, sun angle, and soil moisture. We can add intensity variations caused by the camera to the list. These cause unimportant changes most of the times. On the other hand, we are interested in important changes. Singh recommended the use of images belonging to the same time of the year for reducing the intensity change caused by the sun angle differences and vegetation phenology change. Accurate image registration is also necessary before using satellite images for change detection. Using images without registration can lead to false alarms. In his survey paper, Singh quantitatively evaluated the change detection methods. He concluded that, image regression produced the highest change detection accuracy followed by image ratioing and image differencing. Simple techniques such as image differencing performed better than much more sophisticated transforms such as PCA.

Mas [2] in his survey article compared six change detection methods in terms of land cover change. He focused on a tropical area which is subject to forest clearing. Here, land cover can be classified based on the spectral reflectance of the vegetation area. Mas pointed out that, classification based on the spectral reflectance is difficult for the areas where vegetation diversity is high (such as humid tropics). Therefore, change between land cover types (presenting similar spectral signatures) is difficult to detect. As we referred previously, we can reduce the spectral change caused by the sun angle differences and vegetation phenology change by using images from the same time of the year. Mas indicated that it is extremely difficult to obtain multi-date images taken exactly at the same time of the year, particularly in tropical regions where cloud cover is common. Therefore, he compared the performance of different change detection techniques using images captured at different times of the year.

Accurate image registration is vital before using the multi-temporal images for change detection. In addition to geometrical rectification, images should also be comparable in terms of radiometric characteristics. Mas referred to two ways to achieve radiometric



compensation: radiometric calibration (converting images from digital number values into ground reflection values) and relative radiometric normalization between the multi-temporal images. Mas found that, radiometric normalization is sufficient for change detection. In relative normalization, one image is normalized using the statistical parameters of the other image.

Mas grouped change detection methods under three categories as: image enhancement, multi-date data classification, and comparison of two land cover classifications. He explained each category as: “The enhancement approach involves the mathematical combination of imagery from different dates such as subtraction of bands, rationing, image regression, and PCA. Thresholds are applied to the enhanced image to isolate the pixels that have changed. The direct multi-date classification is based on the single analysis of a combined dataset of two or more different dates, in order to identify areas of changes. The post-classification comparison is a comparative analysis of images obtained at different moments after previous independent classification”. He compared the following six methods on the test area: image differencing, vegetation index differencing, selective PCA, direct multi-date classification, post-classification analysis, and combination of image enhancement and post-classification analysis. Mas reported that post-classification comparison produced the highest accuracy. In single band analysis such as single band differencing, Landsat MSS band 2 (red) produced better results compared to Landsat MSS band 4 (infrared). Based on the same band, PCA produced better accuracy than image differencing. Superior performance of the post-classification comparison is attributed to the difficulty in classifying the land cover using the spectral data. Mas indicated that methods that are directly using the spectral data have problems in classifying land cover which has similar spectral signatures. He mentioned that, the use of classification techniques avoids this problem.

Lu *et al.* [3] investigated a wide range of change detection techniques in their recent survey paper. They listed change detection applications which have attracted attention in the remote sensing community so far. These are: land use and land cover change, forest or vegetation change, forest mortality, defoliation and damage assessment, deforestation, regeneration and selective logging, wetland change, forest fire, landscape change, urban

change, environmental change, and other applications such as crop monitoring. Lu *et al.* grouped change detection methods into seven categories. For our application, the most important of these are: algebra, transformation, and classification based change detection. The algebra category includes image differencing, image regression, image ratioing, vegetation index differencing and Change Vector Analysis (CVA). Lu *et al.* listed the advantages and disadvantages of these methods as follows. “These methods (excluding CVA) are relatively simple, straightforward, easy to implement and interpret, but these cannot provide complete matrices of change information. ... One disadvantage of the algebra category is the difficulty in selecting suitable thresholds to identify the changed areas. In this category, two aspects are critical for the change detection results: selecting suitable image bands or vegetation indices and selecting suitable thresholds to identify the changed areas”. Due to the simplicity of the mentioned methods, they only provide the change and no-change information.

Lu *et al.* also addressed the concept of change matrix in the quotation. A change matrix covers a full range of from-to change classification. A common example includes land cover type changes such as from agricultural to urban or from forest to grassland. They considered the PCA, Kauth-Thomas (KT), Gram-Schmidt, and Chi-square transformations under the transformation category. They listed the advantages and disadvantages of these methods as follows. “One advantage of these methods is in reducing data redundancy between bands and emphasizing different information in derived components. However, they cannot provide detailed change matrices and require selection of thresholds to identify changed areas. Another disadvantage is the difficulty in interpreting and labeling the change information on the transformed images”. Their classification category includes post-classification comparison, spectral-temporal combined analysis, expectation-maximization algorithm (EM) based change detection, unsupervised change detection, hybrid change detection, and artificial neural networks. The advantage of these methods is the capability of providing a change matrix. The disadvantage is the need for a qualified and large training sample set for good classification results.

## 1.2. LAYOUT OF THE THESIS

In this thesis study, we investigate several change detection methods. We group them into four categories as: pixel based, transformation based, texture based, and structure based change detection methods. We explain each method in detail (with their references) in the following chapters. Here, we briefly summarize these chapters.

We investigate pixel based change detection methods in Chapter 2. In this chapter, we investigate direct algebraic calculations such as image differencing and image ratioing; image regression which estimates second-date image by use of linear regression; CVA which accepts pixel values as vectors and provides change information based on vector differences and use of pixelwise fuzzy xor operator for change detection. Pixelwise fuzzy xor operator based method is the novel contribution for any change detection problem.

In Chapter 3, we investigate the transformation based change detection methods. In this chapter, we investigate PCA which is a common technique from the field of multivariate statistical analysis; Kauth-Thomas transformation where the transformed data is directly attributed to the analysis of land-cover; vegetation indices and the time-dependent vegetation indices (commonly used in analysis of change in vegetation) and color invariants. Among these methods, color invariants based change detection is a novel adaptation to this field. Time-dependent vegetation indices are improved in this study.

We investigate texture based change detection methods in Chapter 4. In this chapter, we benefit from Gray Level Co-occurrence Matrix (GLCM) features. We also benefit from the entropy of the image windows as another texture feature.

In Chapter 5 we investigate structure based change detection methods. These can be summarized as the use of edge information, gradient magnitude based support regions, matched filtering, mean shift segmentation, use of local features, bipartite graph matching with local features, and shadow information. Among these methods, use of local features, bipartite graph matching with local features and shadow information are novel contributions to change detection for remote sensing.

We test all our change detection methods on 35 panchromatic and multi-spectral Ikonos satellite test image sets. We provide the quantitative comparison results on these images as well as the strengths and weaknesses of each method in Chapter 6. Finally, in Chapter 7 we summarize the conclusions that we have reached in this study.

## 2. PIXEL BASED METHODS

In this chapter, we consider pixel based change detection methods. First, we provide well-known methods in the literature. Then, we propose two novel pixel based change detection methods.

### 2.1. IMAGE DIFFERENCING

In this technique, images of the same area obtained from times  $t_1$  and  $t_2$  are simply subtracted pixel by pixel. Mathematically, the difference image is defined as

$$I_d(x, y) = I_1(x, y) - I_2(x, y) \quad (2.1)$$

where  $I_1$  and  $I_2$  are the images obtained from  $t_1$  and  $t_2$ .  $x, y$  are the coordinates of the pixels. Resulting image  $I_d$  represents the intensity difference of image  $I_1$  from image  $I_2$ . This technique works only if images are registered.

Here for an interpretation of the difference image, we need to recall the quotation from Singh [1]. “The basic premise in using remote sensing data for change detection is that changes in land cover must result in changes in radiance values and changes in radiance due to land cover change must be large with respect to radiance changes caused by other factors”. Based on this principle, we can expect that intensity differences due to land cover change resides at the tails of the difference distribution of the image. Assuming that changes due to land cover are less than changes by other factors, we expect that most of the difference is distributed around the mean. We can illustrate the difference distribution as in Figure 2.1.

For a zero mean difference distribution, we normalize one image according to the probabilistic parameters of the other image as

$$\tilde{I}_2(x, y) = \frac{\sigma_1}{\sigma_2}(I_2(x, y) - \mu_2) + \mu_1 \quad (2.2)$$

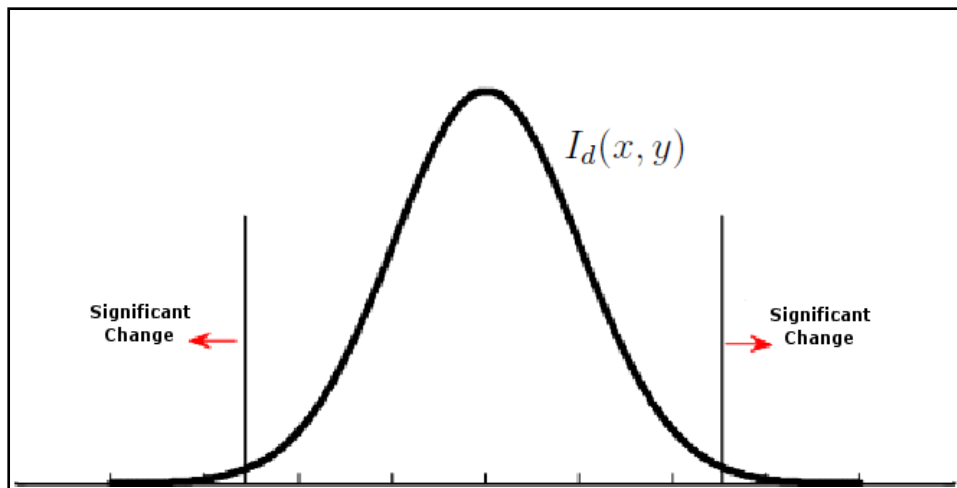


Figure 2.1. Distribution of a difference function. Significant changes are expected at the tails of the distribution

where  $\tilde{I}_2$  is the normalized form of  $I_2$ .  $\mu_1, \sigma_1$  and  $\mu_2, \sigma_2$  are the mean and the standard deviation of  $I_1$  and  $I_2$  respectively. After normalization, the mean and standard deviation of the images are equalized. Hence the difference image will have zero mean. Now, we can update Eqn. 2.1 as

$$I_d(x, y) = |I_1(x, y) - \tilde{I}_2(x, y)| \quad (2.3)$$

To detect the change, we can apply simple thresholding to  $I_d(x, y)$  as

$$T(x, y) = \begin{cases} 1, & I_d(x, y) \geq \tau \\ 0, & \text{otherwise.} \end{cases} \quad (2.4)$$

where the threshold  $\tau$  is often determined empirically.

Since the threshold value in Eqn. 2.1 is important, various automated threshold selection algorithms are proposed. Most of the time, the performance of these algorithms is scene dependent due to the assumptions they are based on. Rosin *et al.* [4] investigated the performance of several automated thresholding algorithms using a large set of difference images calculated from an automatically created ground truth database. They give results

based on several measures for a complete evaluation. In this study, we benefit from three different threshold selection methods. These are percentile thresholding, Otsu's method [5] and Kapur's algorithm [6].

Thresholding methods will be briefly explained next. To this end, we make some assumptions about  $I_d$  as follows.  $I_d$  is a grayscale image which is represented by  $N_g$  gray levels  $\{1, 2, \dots, N_g\}$ . The number of pixels at level  $i$  is denoted by  $n_i$  and the total number of pixels by  $N = n_1 + n_2 + \dots + n_g$ .

The first thresholding method is the percentile. Percentile is a statistics of ordinal scale data. Assume that  $A$  is a sorted array of pixel values of  $I_d$  in ascending order. Rank of the  $P$ -th percentile of  $I_d$  is given by

$$R = \text{ceil}\left(\frac{P}{100} \times N\right) \quad (2.5)$$

where *ceil* function rounds its argument to the nearest greater integer.  $P$ -th percentile is found by indexing  $A$  using that rank.

The second thresholding method is proposed by Otsu. It uses measures of class separability in finding an optimal threshold value. Relative frequencies of pixel values at level  $i$  are given by

$$p_i = \frac{n_i}{N}, \quad p_i \geq 0, \quad \sum_{i=1}^L p_i = 1 \quad (2.6)$$

A threshold value at gray level  $k$  divides the histogram of  $I_d$  into two classes. Each class has its own probability of occurrence (total probability of its samples) and own mean value. Evaluation function of the Otsu's method is the between-class variance given by

$$\sigma_b^2 = \frac{[\mu_{H_{I_d}} \omega(k) - \mu_\omega]^2}{\omega(k)[1 - \omega(k)]} \quad (2.7)$$

where  $\mu_{H_{I_d}}$  is the mean of the histogram of  $I_d$ ;  $\omega(k)$  is the probability of the class which

includes gray levels up to  $k$  and  $\mu_\omega$  is the mean of the class  $\omega$ . Optimal threshold value  $k^*$  maximizes

$$\sigma_b^2(k^*) = \max_{1 \leq k \leq N_g} \sigma_b^2(k) \quad (2.8)$$

The last thresholding method we use is Kapur's algorithm. Similar to the Otsu's method, it divides the image histogram into two classes and then utilizes the sum of the entropy of these two classes as an evaluation function. The value which maximizes this sum is taken as the optimal threshold value. For two classes  $A$  and  $B$ , Shannon entropy of these classes are defined as

$$H(A) = - \sum_{i=1}^k \frac{p_i}{\omega(k)} \ln \frac{p_i}{\omega(k)} \quad (2.9)$$

$$H(B) = - \sum_{i=k+1}^{N_g} \frac{p_i}{[1 - \omega(k)]} \ln \frac{p_i}{[1 - \omega(k)]} \quad (2.10)$$

where the histogram is divided at gray level  $k$ . The optimal threshold value  $k^*$  maximizes the sum

$$\phi(k) = H(A) + H(B), \quad \phi(k^*) = \max_{1 \leq k \leq N_g} \phi(k) \quad (2.11)$$

To explain different change detection methods, we pick the Adana test image set given in Figure 2.2. The two images, taken in different times, in this set represent a region with construction activity. These images are registered. Therefore, they can be used for pixelwise change detection methods. The difference between these two images is clearly seen. We will use this image set in the following sections also.

The difference image obtained from the Adana image set is as in Figure 2.3.a. This image is color coded with the color scale given next to it. We also provide the thresholding result in Figure 2.3.b. In thresholding, we benefit from Kapur's method.





Figure 2.2. Images taken at two different times from a developing region of Adana.

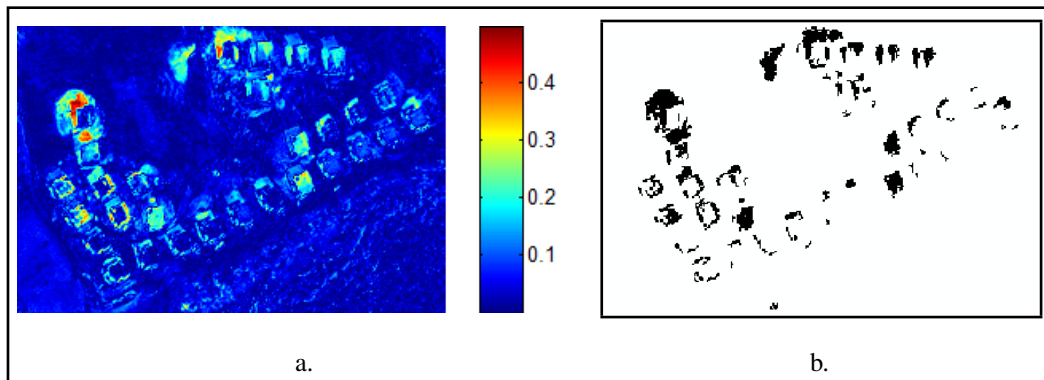


Figure 2.3. Image differencing applied to the Adana image set: a. The difference image; b. Thresholded version

Griffiths [7] used image differencing for estimating the urban change. He used Landsat TM (30 m resolution) data, SPOT XS (20 m resolution) multispectral data and SPOT panchromatic (10 m resolution) data in his study. He proposed some refinements to image differencing for the purpose of finding urban change. An urban mask is produced from the SPOT panchromatic data. Griffiths indicated that mix of buildings, streets and small gardens in urban areas produces a highly textured appearance compared with the much smoother texture of arable fields. He used a standard deviation filter to quantify the texture. Urban mask was multiplied by the difference image to eliminate non-urban areas. Furthermore, he refined the results based on a previous study. In this technique, changes that occur far from the urban areas are assumed to be non-urban change. This is because new urban development generally occurs at the periphery of existing urban areas. Griffiths presented his results for each technique by comparing them with visual interpretation.

Saksa *et al.* [8] used image differencing for detecting clear cut areas in boreal forests. They tested three methods using Landsat satellite imagery and aerial photographs as: pixel-by-pixel differencing and segmentation, pixel block-level differencing and thresholding, and pre-segmentation and unsupervised classification. In the first method, they found the difference image. Then they used a segmentation algorithm to delineate the clear cut areas. In the second method, they included neighboring pixels into the calculation of the difference image. Therefore, negative effects of misregistration is reduced in the resulting image. In the third method, they first segmented the images. Then, they obtained a segment-level image difference. They labeled clear cut areas by using an unsupervised classification algorithm. Saksa *et al.* concluded that predelineated segments or pixel blocks should be used for image differencing in order to decrease the amount of misinterpreted small areas.

Lu *et al.* [9] compared 10 binary change detection methods to detect land cover change in Amazon tropical regions. They used TM (Thematic Mapper) data in their study. In addition to band differencing, they tested a modified version of image differencing where pixels are accepted as changed when majority of the bands indicate change. For six-band TM data, if four of the bands indicate change then the pixel value is labeled as changed. They reported that difference of TM band 5, modified image differencing and principal component differencing produced best results.

## 2.2. IMAGE RATIOING

In this method, images are compared pixel by pixel as in the previous method. Similar to image differencing, images must be registered before applying image rationing. The ratio image is calculated by

$$I_r(x, y) = \frac{I_1(x, y)}{\tilde{I}_2(x, y)} \quad (2.12)$$

In Eqn. 2.12, the  $I_r$  image takes values in the range  $[0, \infty)$ . If the intensity values are equal, it takes the value 1. To normalize the value of  $I_r$ , we can benefit from the arctangent function as

$$I_r(x, y) = \arctan \left( \frac{I_1(x, y)}{\tilde{I}_2(x, y)} \right) - \frac{\pi}{4} \quad (2.13)$$

Now, ratio image takes values in the range  $[-\pi/4, \pi/4]$ . To threshold  $I_r$ , we can benefit from the same methods as we did in the previous section. In Figure 2.4.a, we provide the  $I_r$  image obtained from the Adana test image set. We provide the thresholded version of this image in Figure 2.4.b. As in the previous section, we used Kapur's method to obtain the optimal threshold value.

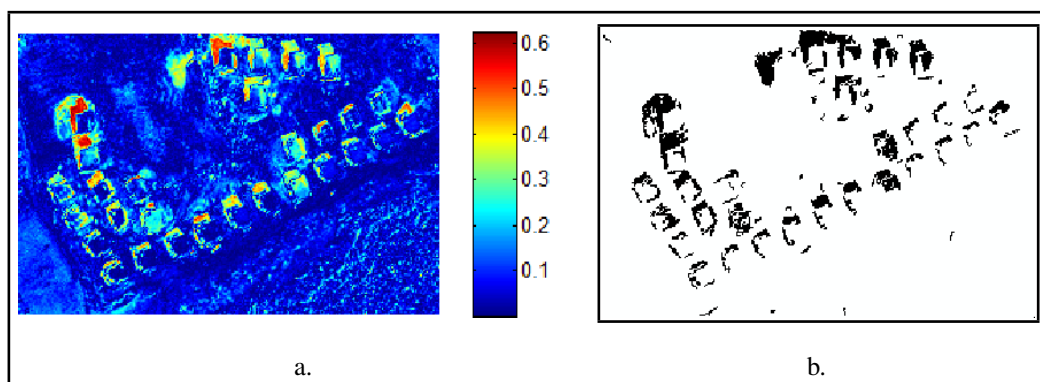


Figure 2.4. Image ratio applied to the Adana image set: a. The ratio image; b. Thresholded version

### 2.3. IMAGE REGRESSION

In image regression, the  $I_2$  image (obtained from  $t_2$ ) is assumed to be a linear function of the  $I_1$  image (obtained from  $t_1$ ). Under this assumption, we can find an estimate of  $I_2$  by using least-squares regression as

$$\hat{I}_2(x, y) = b + aI_1(x, y) \quad (2.14)$$

To estimate the parameters  $a$  and  $b$ , we define the squared error between the measured data and predicted data (for each pixel) as

$$e^2 = (I_2(x, y) - \hat{I}_2(x, y))^2 = (I_2(x, y) - b - aI_1(x, y))^2 \quad (2.15)$$

The sum of the squared error becomes

$$S = \sum e^2 = \sum (I_2(x, y) - b - aI_1(x, y))^2 \quad (2.16)$$

Here, we assume that we have  $N$  observations. We want to find the parameters  $a$  and  $b$  to minimize the sum of the squared error  $S$ . Therefore, we first the partial derivatives of  $S$  with respect to  $a$  and  $b$  as

$$\frac{\partial S}{\partial b} = -2 \sum (I_2(x, y) - b - aI_1(x, y)) \quad (2.17)$$

$$\frac{\partial S}{\partial a} = -2 \sum [(I_2(x, y) - b - aI_1(x, y))I_1(x, y)] \quad (2.18)$$

By equating Eqn. 2.17 and 2.18 to zero, we obtain two equations with two unknowns as

$$0 = \sum I_2(x, y) - \sum b - \sum aI_1(x, y) \quad (2.19)$$

$$0 = \sum I_2(x, y)I_1(x, y) - \sum bI_1(x, y) - \sum aI_1(x, y)^2 \quad (2.20)$$

Solving these equations, we obtain

$$a = \frac{n \sum_{n=1}^N I_2(x_n, y_n) I_1(x_n, y_n) - \sum_{n=1}^N I_2(x_n, y_n) \sum_{n=1}^N I_1(x_n, y_n)}{n \sum_{n=1}^N I_1(x_n, y_n)^2 - (\sum_{n=1}^N I_1(x_n, y_n))^2} \quad (2.21)$$

$$b = \frac{\sum_{n=1}^N I_2(x_n, y_n) - a \sum_{n=1}^N I_1(x_n, y_n)}{N} \quad (2.22)$$

We manually picked the observations (for  $n = 1, \dots, N$ ) from  $I_1$  and  $I_2$  (from the unchanged areas). As a result, the estimate of  $I_2$  carries unimportant differences due to factors referred in the image differencing section. When we subtract  $I_2$  from  $\hat{I}_2$  as  $I_d(x, y) = I_2(x, y) - \hat{I}_2(x, y)$ , we expect to find the changes originating from land cover. When we apply this method to the normalized  $I_2$  ( $\tilde{I}_2$ ), we further eliminate the insignificant changes that still remain after normalization. Consequently, this technique gives slightly better performance than image differencing.

We provide the difference image obtained by image regression using our Adana image test set in Figure 2.5.a. We provide the thresholded version of this image in Figure 2.5.b. As in the previous sections, we benefit from Kapur's method in threshold selection.

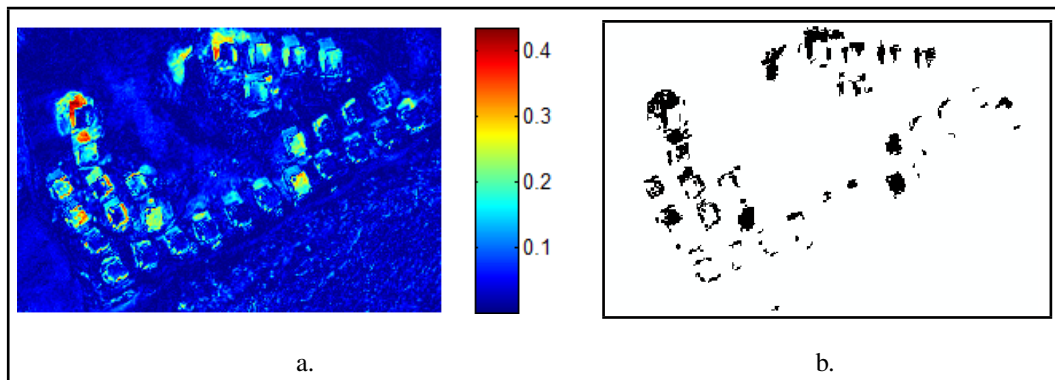


Figure 2.5. Image difference after regression applied to the Adana image set: a. The difference image; b. Thresholded version

## 2.4. CHANGE VECTOR ANALYSIS

Change Vector Analysis (CVA) is a change analysis technique where multiple image bands can be analyzed simultaneously. As its name suggests, CVA does not only function as a change detection method but also helps analyzing and classifying change. In CVA, pixel values are vectors of spectral bands. Change vectors (CV) are calculated by subtracting vectors pixel by pixel as in image differencing. Magnitude and direction of the change vectors are used for change analysis. In Figure 2.6 a changed pixel and an unchanged pixel are given in a two-band spectral space.

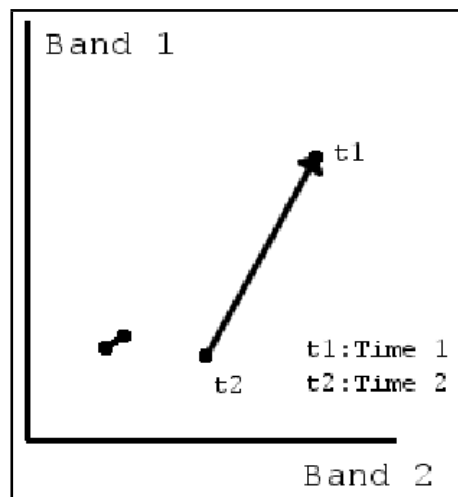


Figure 2.6. Unchanged and changed pixel vectors in a 2-D spectral space

Change vector magnitude can be used as a measure of change and thus for change and no-change classification. Under ideal conditions, such as perfect image registration and normalization, unchanged pixel magnitudes must be equal to zero. However, this is not the case in practical applications. Therefore, thresholding must be applied to the change magnitude. While change vector magnitude behaves like a multi-band version of the image differencing, change direction gives us information about the type of change. This information is often more valuable than the amount of change, since in most applications we are interested in a specific type of change.

In application, the number of CV directions are uncountable. Therefore, it is necessary to divide the CV space into subsets and assign directions accordingly. A simple discretization

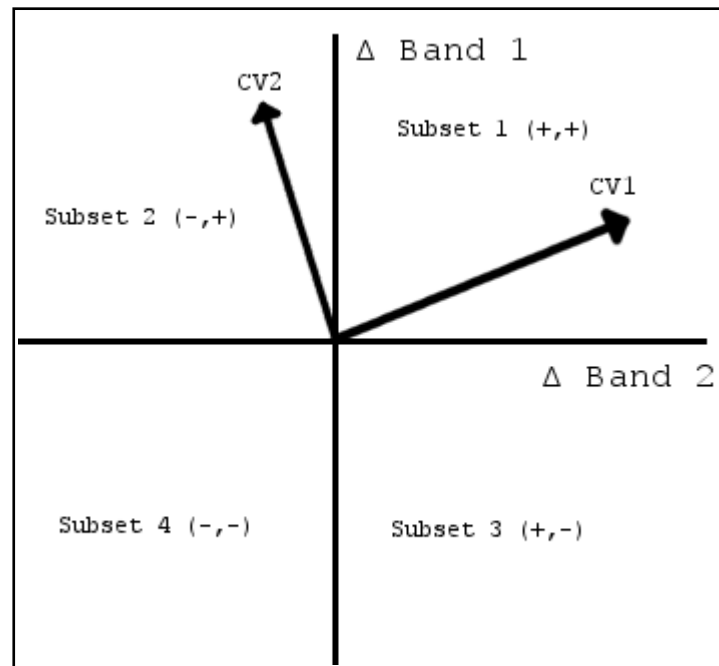


Figure 2.7. Change vector space divided into four subsets

of CV directions can be achieved by dividing the space by its main axes. In Figure 2.7 a two dimensional CV space is divided into four subsets (quadrants) by the axis of band 1 and band 2. For three band images, subsets can be octants. CVs can be assigned to subsets via signs of their components. For an  $n$ -D space components, the CVs can take  $2^n$  different sign combinations so that there exists  $2^n$  subsets.

As mentioned earlier, CV directions can be used in classifying the change. By using subsets, we can determine  $2^n$  classes of change for an  $n$  dimensional space. CVA can also be applied to transformed data such as Kauth-Thomas Transformation (KTT) (explained in Section 3.2) rather than to raw data. For instance from KTT space, a simultaneous increase in the greenness feature and decrease in the brightness feature indicates gain of vegetation. Therefore, in the change vector space of KTT bands, we can assign this class of change (change towards vegetation) to the subsets where greenness is positive and brightness is negative.

CVA for change detection is introduced by Malila [10]. Malila used the KTT with CVA and reported results for change of forestation. Change directions were used to

distinguish between changes due to harvesting and changes due to regrowth. Johnson *et al.* [11] provided a comprehensive investigation of CVA. They provided the details to the implementation of CVA after a functional description. They reported that, CVA can be used in applications which require a full-dimensional data processing and analysis technique and require capturing all changes. They also found CVA to be useful for applications in which: the changes of interest and their spectral manifestation are not well known a priori; changes of interest are known or thought to have high spectral variability; and/or changes in both land cover type and condition may be of interest.

In Figure 2.8.a, we provide the change vector magnitude image which can be used for change and no-change classification. In Figure 2.8.b the change vector magnitude image is thresholded by Kapur's algorithm.

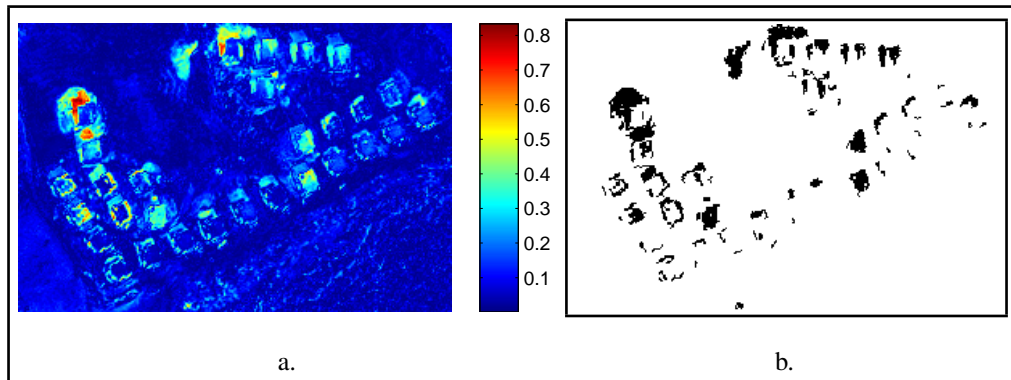


Figure 2.8. CVA (in terms of magnitude value) applied to the Adana image set: a. The magnitude image; b. Thresholded version

## 2.5. PIXELWISE FUZZY XOR OPERATOR

The last method for pixelwise change detection is a novel contribution to the community. In this method, the binary XOR operation is taken as a benchmark and its fuzzy version is used for change detection. Our rationale is as follows. Assume that we have two binary images (composed of only ones and zeros) and we want to detect the changed pixels in these images. Each pixel  $p(x, y)$  in a binary image  $B$  is valued according to a characteristic function  $\beta_B$ ,



which could also be called the “whiteness” function defined as

$$p(x, y) = \beta_B(x, y) = \begin{cases} 1, & \text{if } B(x, y) \text{ is white} \\ 0, & \text{otherwise} \end{cases} \quad (2.23)$$

Between two pixels  $p_1$  and  $p_2$  at the same  $(x, y)$  coordinates of the two binary images  $B_1$  and  $B_2$ , the existence of a change can only mean that either “ $p_1$  is white and  $p_2$  is not” or “ $p_1$  is not white and  $p_2$  is”. This wording directly implies the XOR operation in binary logic. Hence the obvious solution to the change detection problem is XOR-ing the two binary images pixel-by-pixel, defined as

$$C(x, y) = B_1(x, y) \oplus B_2(x, y) \quad (2.24)$$

This operation gives ‘0’ for pixels having the same values in both images, and gives ‘1’ for pixels having different values. Therefore, white pixels in the resulting binary image  $C(x, y)$  represent the changed regions.

Unfortunately, this method cannot be applied to panchromatic or multispectral satellite imagery (having pixel values in a certain range). In order to perform a similar change detection on satellite imagery, we propose a fuzzy representation for the images and a corresponding change detection method on a combination of fuzzy and crisp (binary) operations.

Panchromatic images are composed of pixels with values  $p(x, y)$  in a certain range. Normalizing these values and mapping them to the range  $[0, 1]$  effectively translates the image into a fuzzy set, whose elements (pixels) have membership grades in proportion to their “whiteness”. The membership grade  $g(x, y)$  of each pixel  $p(x, y)$  in the grayscale

image  $G$  is thus defined by the fuzzy membership function  $\mu_G$  as

$$g(x, y) = \mu_G(x, y) = \begin{cases} 1.00, & \text{if } G(x, y) \text{ is pure white} \\ \dots & \dots \\ 0.50, & \text{if } G(x, y) \text{ is gray} \\ \dots & \dots \\ 0.00, & \text{if } G(x, y) \text{ is pure black} \end{cases} \quad (2.25)$$

Comparison of two binary images involves the crisp question “Are these two pixels different?”. Whereas a fuzzy comparison of two panchromatic images involves the fuzzy question “How different are these two pixels?”. Also the question of “Above what amount of difference shall the two pixels be labeled as changed?”. The amount of difference between gray level values in the image domain directly corresponds to the difference between the degrees of membership in the fuzzy domain. For this particular application, the fuzzy complement (NOT) operation, defined as

$$\bar{g}(x, y) = \mu_{\bar{G}}(x, y) = 1 - g(x, y) \quad (2.26)$$

and the algebraic representation of the fuzzy intersection (AND) operation, defined as the multiplication of membership functions

$$\mu_{G_1 \cap G_2} = \mu_{G_1}(x, y) \mu_{G_2}(x, y) = g_1(x, y) g_2(x, y) \quad (2.27)$$

were used to obtain a fuzzy difference metric [12].

In a manner similar to the binary case, the measure of change between two pixels  $p_1$  and  $p_2$  is given by the degree of truth of the following statement: either “ $p_1$  is lighter AND  $p_2$  is darker” OR “ $p_1$  is darker AND  $p_2$  is lighter”; which can be rephrased as, either “ $p_1$  has a high membership grade AND  $p_2$  has a low membership grade” OR “ $p_1$  has a low membership grade AND  $p_2$  has a high membership grade”.

Considering that “having a low membership grade” is the opposite of “having a high membership grade”, the former statement’s degree of truth is the complement of the latter’s, and the degree of truth in “having a high membership grade” is equivalent to the membership grade  $g(x, y)$  itself. Consequently, the above fuzzy rule can be formulated as

$$C(x, y) = \mu_{(G_1 \cap \bar{G}_2) \cup (\bar{G}_1 \cap G_2)}(x, y) = (g_1(x, y)\bar{g}_2(x, y)) \cup (\bar{g}_1(x, y)g_2(x, y)) \quad (2.28)$$

The fuzzy value  $C(x, y)$  represents the measure of change between two images at coordinates  $(x, y)$ . The decision of a significant change can be made by means of applying an appropriate threshold and converting  $C(x, y)$  to a crisp YES/NO value. Experiments have shown that, the results from the two fuzzy AND operations are distributed in a way that automatically indicates an appropriate threshold for defuzzification. More explicitly, threshold values are obtained for both fuzzy AND operations from  $\tau = \text{argmax}(H_a) + 2\sigma_a$ . Here,  $H_a$  is the histogram of the corresponding fuzzy AND operation and  $\sigma_a$  is the standard deviation of the corresponding fuzzy AND operation. In fact, applying this threshold and converting the fuzzy AND results to a crisp binary value, and then combining them with the binary OR operator yielded better results in detecting changed regions in satellite images. Therefore, the proposed method was eventually established as an ensemble of both fuzzy and binary logic operations.

We provide the images obtained by fuzzy AND operations using our Adana images in Figure 2.9.a and Figure 2.9.b. We provide the thresholded version after finding the  $C(x, y)$  in Figure 2.9.c.

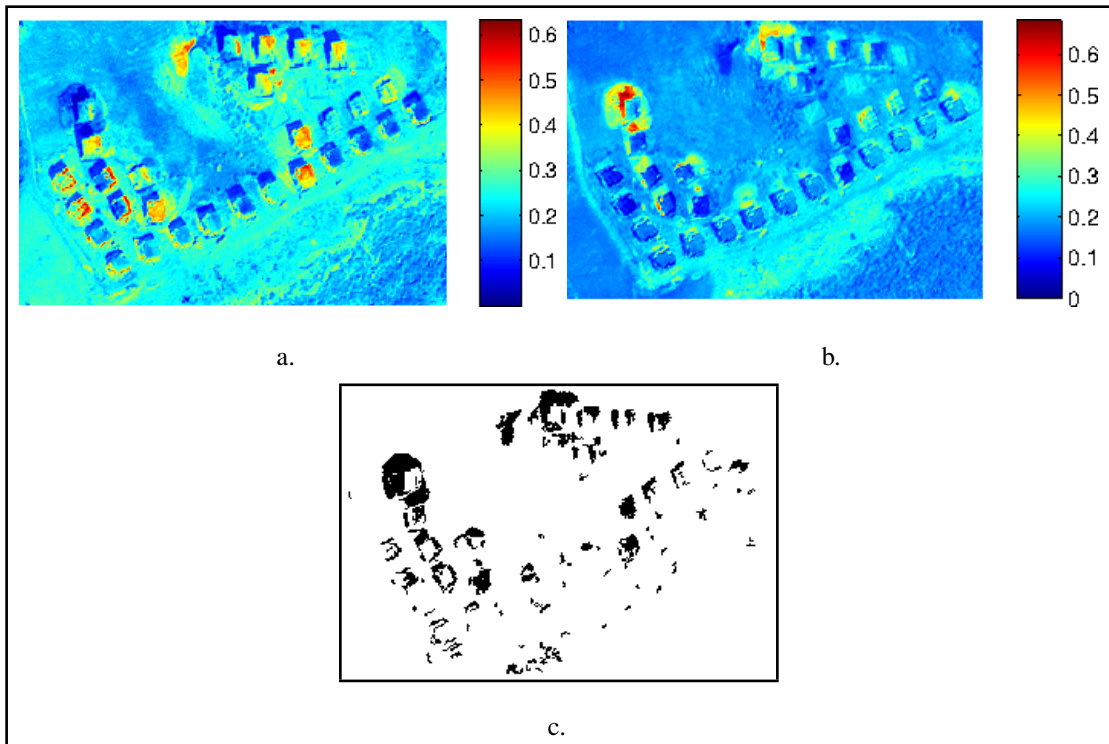


Figure 2.9. Fuzzy XOR applied to the Adana image set: a. Fuzzy AND ( $g_1(x, y)g_2(x, y)$ );  
 b. Fuzzy AND ( $\bar{g}_1(x, y)g_2(x, y)$ ); c. Thresholded version

### 3. TRANSFORMATION BASED METHODS

This chapter deals with change detection methods based on color or multispectral space transformations. They are based on Principal Component Analysis, Kauth-Thomas transformation, and vegetation indices.

#### 3.1. PRINCIPAL COMPONENT ANALYSIS (PCA)

PCA is the transformation of the multivariate data to a new set of components where data variation can be expressed by a few first components. PCA achieves this by removing the redundancy in the data set. This redundancy is quantified by the correlation of the variables, and PCA transforms a correlated set of data to an uncorrelated set.

In terms of linear algebra what PCA does is a basis rotation. The basis rotation can be defined in an algorithmic manner as follows. Variance of the projections onto the first basis vector ( $e_1$ ) takes its maximum after the rotation. Under the assumption that  $e_1$  is fixed (e.g. rotation axis is  $e_1$ ), variance of the projections onto the second basis vector ( $e_2$ ) takes its possible maximum after the rotation. Variance of the projections onto the basis vector  $e_m$  takes its possible maximum under the assumption that vectors from the previous steps ( $e_1, e_2, \dots, e_{m-1}$ ) are fixed. Data is redefined under a new basis ( $e_1, e_2, \dots, e_n$ ).

PCA is algebraically defined as follows. The sample covariance of  $N$  observations of  $K$  variables ( $X_1, X_2, \dots, X_K$ ) is the  $K$  by  $K$  matrix  $C_x = [c_{jk}]$  with the entries

$$c_{jk} = \frac{1}{N-1} \sum_{i=1}^N (x_{ij} - \mu_j)(x_{ik} - \mu_k) \quad (3.1)$$

where  $x_{ij}$  is the  $i^{th}$  observation of the  $j^{th}$  variable.  $\mu_j$  and  $\mu_k$  are the mean of  $j^{th}$  and  $k^{th}$  variables respectively. Based on these, the PCA transformation can be defined as  $\mathbf{Y} = \mathbf{XU}$  where  $\mathbf{U}$  is the  $K$  by  $K$  rotation matrix whose columns are the eigenvectors of  $C_x$ .  $\mathbf{X}$  is the  $N$  by  $K$  data matrix whose columns represent the variables and its rows represent the

observations. Columns of  $Y$  are Principal Components (PCs). Correlation of the variables to the PCs is a special measure and named as principal component loadings. The principal component loadings indicate how much variance in each of the variables is accounted for by the PCs.

Application of the PCA to change detection requires the analysis of the PC loadings. There exist two approaches to analyze the multi-temporal images in the context of change detection. The first approach is called *separate rotation*. In this approach, the PCA is applied to multi-band images separately. Then, any of the change detection techniques such as image differencing is applied to the PCs. The second approach is called *merged rotation*. In this approach, data from the bitemporal images are merged into one set and PCA is applied to it. PCs which account for the change are selected via analysis of the PC loadings. These PCs have negative correlation to the bitemporal data. Negative correlation is found in PCs whose loadings are positive to one image and negative to the other image.

Fung and LeDrew [13] applied PCA to land cover change detection. They calculated the eigenvectors from the correlation matrix and covariance matrix, and compared the change detection performance when PCA is applied using each eigenvector. They reported that PCA with eigenvectors calculated from the correlation matrix gives better results. They first reported results from the separate rotation of the multi-temporal data. They indicated that a careful examination of the PC loadings is necessary before applying change detection to PCs found after separate rotation. Second, they analyzed the results from the merged rotation. They listed the PCs which are responsible for the change in terms of brightness and greenness. They reported that PC loadings from the correlation matrix are better aligned compared to PC loadings from the covariance matrix. In applying the PCA to our test images, we observed that the separate rotation approach has given better results compared to the merged rotation approach.

We used multi-spectral images in our PCA application. Therefore, we benefit from the near infrared band as well as visible bands red, green, and blue. Since multi-spectral images have lower resolution compared to panchromatic counterparts, results are not as rich as panchromatic images in terms of visual interpretation. Still PCA gives good

results compared to the pixel based methods. In Figure 3.1, we provide the differences of principal components for the Adana image set. As in the previous sections, we benefit from Kapur's method in threshold selection. As can be seen, the difference of the third principal components emphasized the changes fairly well.

### 3.2. KAUTH-THOMAS TRANSFORMATION

Kauth-Thomas transformation (KTT) is a linear transformation from the multi-spectral data space to a new space where dimensions are directly attributed to analyze the land cover [14]. In fact, similar to PCA, KTT is a redefinition of the data. Different from PCA, KTT is a fixed transformation and is described as

$$\begin{bmatrix} br \\ gr \\ ye \\ ns \end{bmatrix} = \begin{bmatrix} 0.433 & 0.632 & 0.586 & 0.264 \\ -0.290 & -0.562 & 0.6 & 0.491 \\ -0.829 & 0.522 & -0.039 & 0.194 \\ 0.223 & 0.012 & -0.543 & 0.810 \end{bmatrix} \begin{bmatrix} g \\ r \\ n1 \\ n2 \end{bmatrix} \quad (3.2)$$

where  $X$  is the multi-spectral data matrix. In the original work,  $X$  consists of data from the green( $g$ ), red( $r$ ) and two near infrared ( $n1$ ,  $n2$ ) bands of the Landsat Multi Spectral Scanner (MSS) [14]. KTT features are brightness, greenness, yellowness and non-such.

Fixed character of this transformation is explained by the invariant nature of the correlation between the visible bands and the near infrared bands. While the correlation between the visible bands and the correlation between the near infrared bands are always high, the correlation between the visible bands and the near infrared bands is always low. Kauth and Thomas described this character by visualizing the four band data via their principal components. They had a three dimensional representation of the data which resembles a tasseled cap. Hence, this transformation is also known as the Tasseled Cap transformation. By this character, KTT is scene independent and often referred as a better choice against PCA.

Kauth and Thomas [14] used this transformation to describe the lifecycle of the croplands.

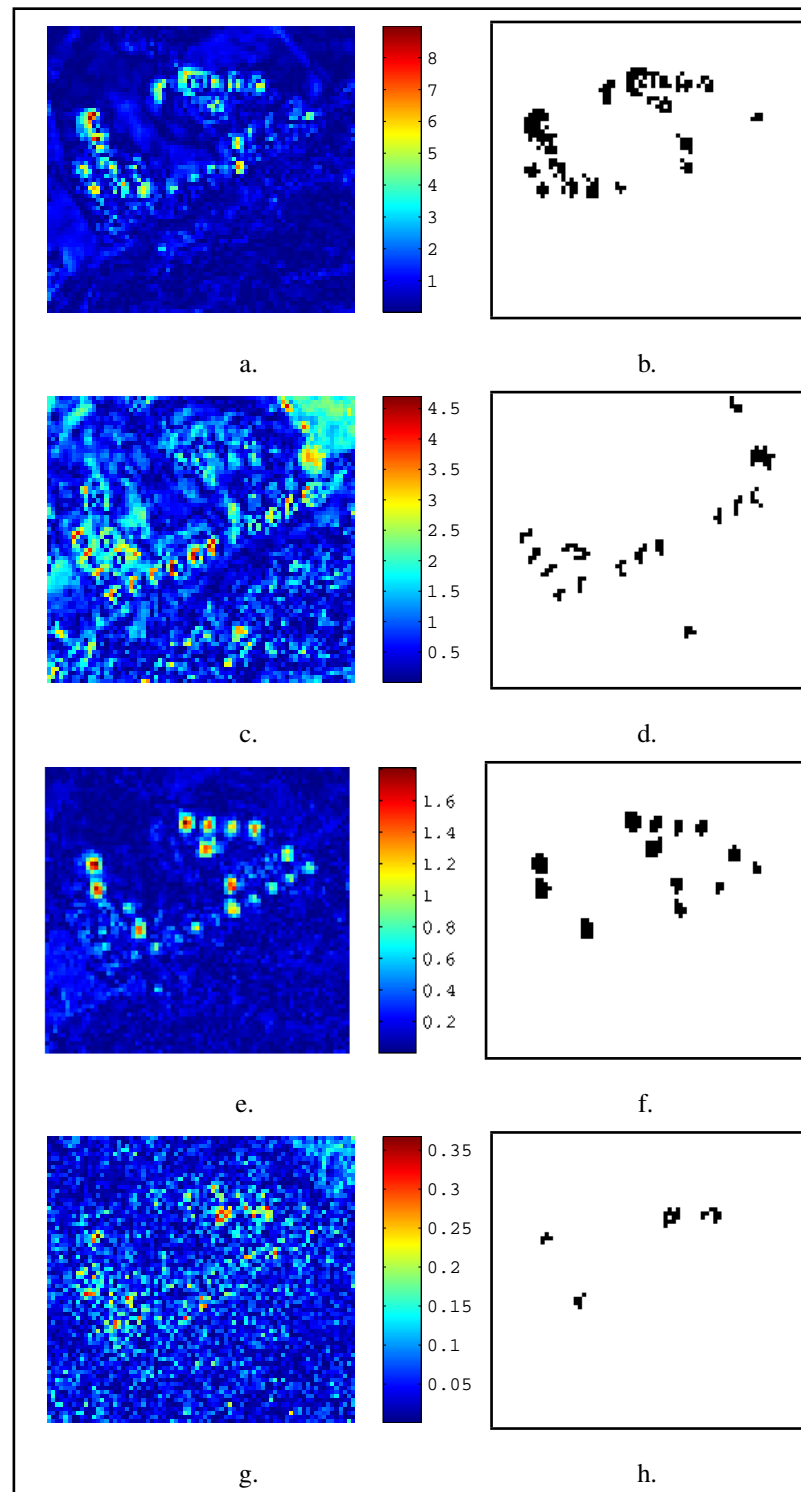


Figure 3.1. Difference of principal components for the Adana image set: a. Difference of the first PCs; c. Difference of the second PCs; e. Difference of the third PCs; g. Difference of the fourth PCs; b.,d.,f. and h. Thresholded versions



They used the brightness feature to find the soil where crop grows on. Increase in the greenness feature indicates the growth of crop until it matures. At the end of the lifecycle, the crop reaches to senescent stage and yellowness feature increases in parallel.

Seto *et al.* [15] applied the Kauth-Thomas transformation for land use change detection in a fast developing area, The Pearl River Delta in the Peoples Republic of China. They referred to the direct association between the physical scene attributes and KTT bands and found that KTT space is easily comprehensible. Land cover types such as forest and urban are determined by the spectral locations in the KTT space (e.g. amount of brightness, greenness and yellowness). Land use change, such as agricultural to urban, is classified based on the change from one land cover type to another.

In applying KTT to our test images, we first need a transformation matrix convenient for Ikonos images. As defined earlier, KTT is applied to the data from green, red and two near infrared bands. However, in Ikonos images we have only one near infrared band. In order to use KTT matrix with green, red and one near infrared band, we first remove the fourth row and fourth column of the matrix. Since resulting matrix is not orthogonal, we applied Gram-Schmidt orthogonalization [16] and obtain

$$\begin{bmatrix} br \\ gr \\ ye \end{bmatrix} = \begin{bmatrix} 0.4489 & 0.6552 & 0.6076 \\ -0.2669 & -0.5506 & 0.7910 \\ -0.8528 & 0.5172 & 0.0723 \end{bmatrix} \begin{bmatrix} g \\ r \\ n \end{bmatrix} \quad (3.3)$$

We provide the differences of brightness, greenness and yellowness bands for the Adana image set in Figure 3.2. As in the previous sections, we used Kapur's algorithm in finding the threshold value. As can be seen, the brightness and yellowness bands indicate changed regions.

### 3.3. VEGETATION INDEX DIFFERENCING

Vegetation indices are obtained by transforming the data from the near infrared and red bands of the multi-spectral data. They are used as a measure of vegetation which depends

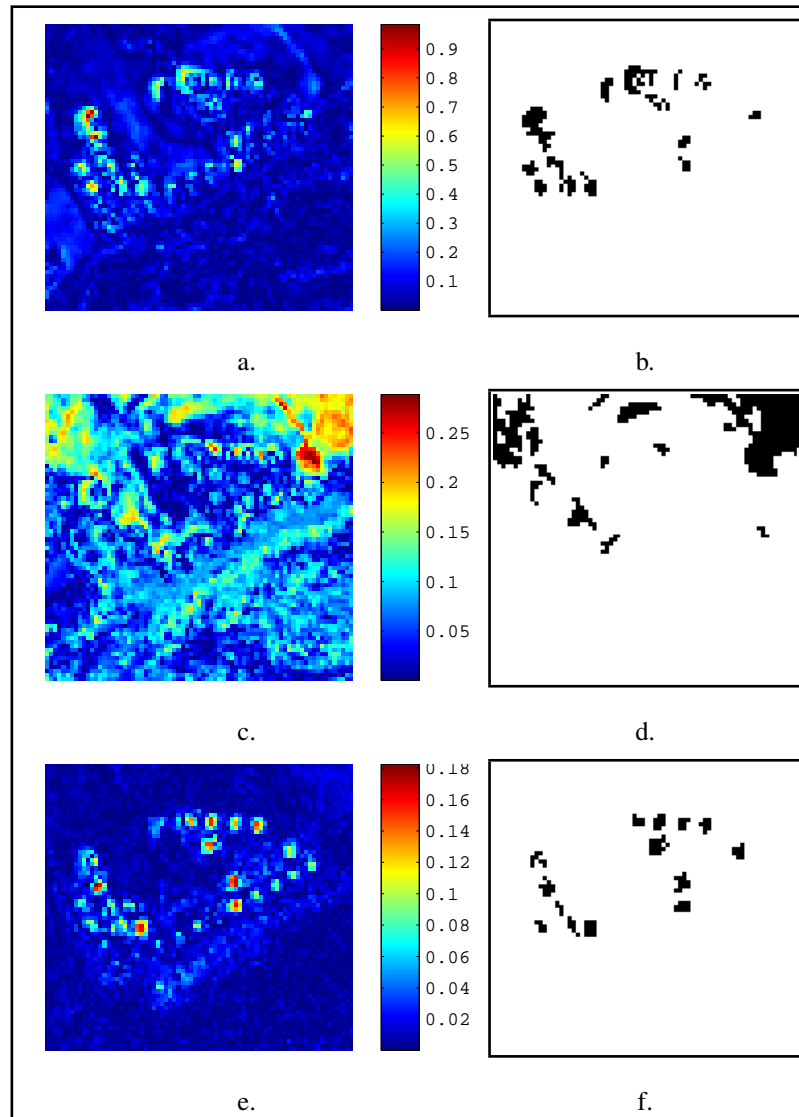


Figure 3.2. Difference of KTT bands for the Adana image set: a. Difference of the brightness; c. Difference of the greenness; e. Difference of the yellowness; b.,d. and f.

Thresholded versions

on the fact that vegetation absorbs most of the light in the red band and equally reflects in the infrared band. There are several types of vegetation indices such as Ratio Vegetation Index (RVI), Normalized Difference Vegetation Index (NDVI), Transformed Vegetation Index (TVI), Soil Adjusted Vegetation Index (SAVI), and Modified Soil Adjusted Vegetation Index (MSAVI). These indices are defined as

$$RVI = \frac{n}{r} \quad (3.4)$$

$$NDVI = \frac{n - r}{n + r} \quad (3.5)$$

$$TVI = \sqrt{\frac{n - r}{n + r} + 0.5} \quad (3.6)$$

$$SAVI = \frac{n - r}{n - r + L}(1 + L) \quad (3.7)$$

$$MSAVI = \frac{2n + 1 - \sqrt{(2n + 1)^2 - 8(n - r)}}{2} \quad (3.8)$$

where  $n$  is the near infrared band and  $r$  is the red band.  $L$  in SAVI confirms the same bounds between NDVI and SAVI.

RVI is an earlier attempt for explaining vegetation by band ratios and introduced in [17]. NDVI and TVI are proposed as an alternative to RVI [18]. Jackson and Huete [19] showed that NDVI is more sensitive to sparse vegetation compared to RVI, but less sensitive to dense vegetation. Lautenschlager and Perry [20] mathematically showed that RVI and NDVI are highly correlated and thus contains the same information. Huete introduced the SAVI to minimize soil brightness influences from spectral vegetation indices involving red and near-infrared wavelengths [21]. He studied on cotton and grassland canopies and showed that SAVI can eliminate soil originating variations in vegetation indices. MSAVI is defined in [22].

In terms of change detection, vegetation indices can be used for measuring alteration of

a vegetation area in time. Any of the pixel based methods described in this study can be applied using vegetation indices for an estimation of change in vegetation.

Lunetta *et al.* [23] investigated the applicability of high resolution NDVI (250 m), MODIS NDVI, to land-cover change detection. They studied on a geographic area where biological diversity and regrowth rates are high. Their results indicate up to 87% correct change detection rates. Guerra *et al.* [24] used the MSAVI with bi-temporal Landsat TM (Thematic Mapper) images for identification of vegetation changes. Guerra *et al.* distinguished six types of land-cover from a tropical area based on the spectral locations in the MSAVI space. They applied normalized image differencing (Eq. 3.9) for quantifying change as

$$D = \frac{MSAVI_{t_2} - MSAVI_{t_1}}{MSAVI_{t_2} + MSAVI_{t_1}} \quad (3.9)$$

In Figure 3.3, we provide the difference images of RVI, NDVI, TVI and SAVI for our Adana image set. Unfortunately, none of the indices provided useful results for change detection on the Adana image set.

### 3.4. TIME DEPENDENT VEGETATION INDEX

A Time Dependent Vegetation Index (TDVI) is a bi-temporal vegetation index which calculated by using multi-spectral bands from  $t_1$  and  $t_2$  [25]. Red and near infrared bands from different times are involved in the same index formula. In the original paper, Ünsalan used angle vegetation indices [26] as

$$\psi = \frac{4}{\pi} \arctan \left( \frac{n}{r} \right) \quad (3.10)$$

$$\theta = \frac{4}{\pi} \arctan \left( \frac{n - r}{n + r} \right) \quad (3.11)$$

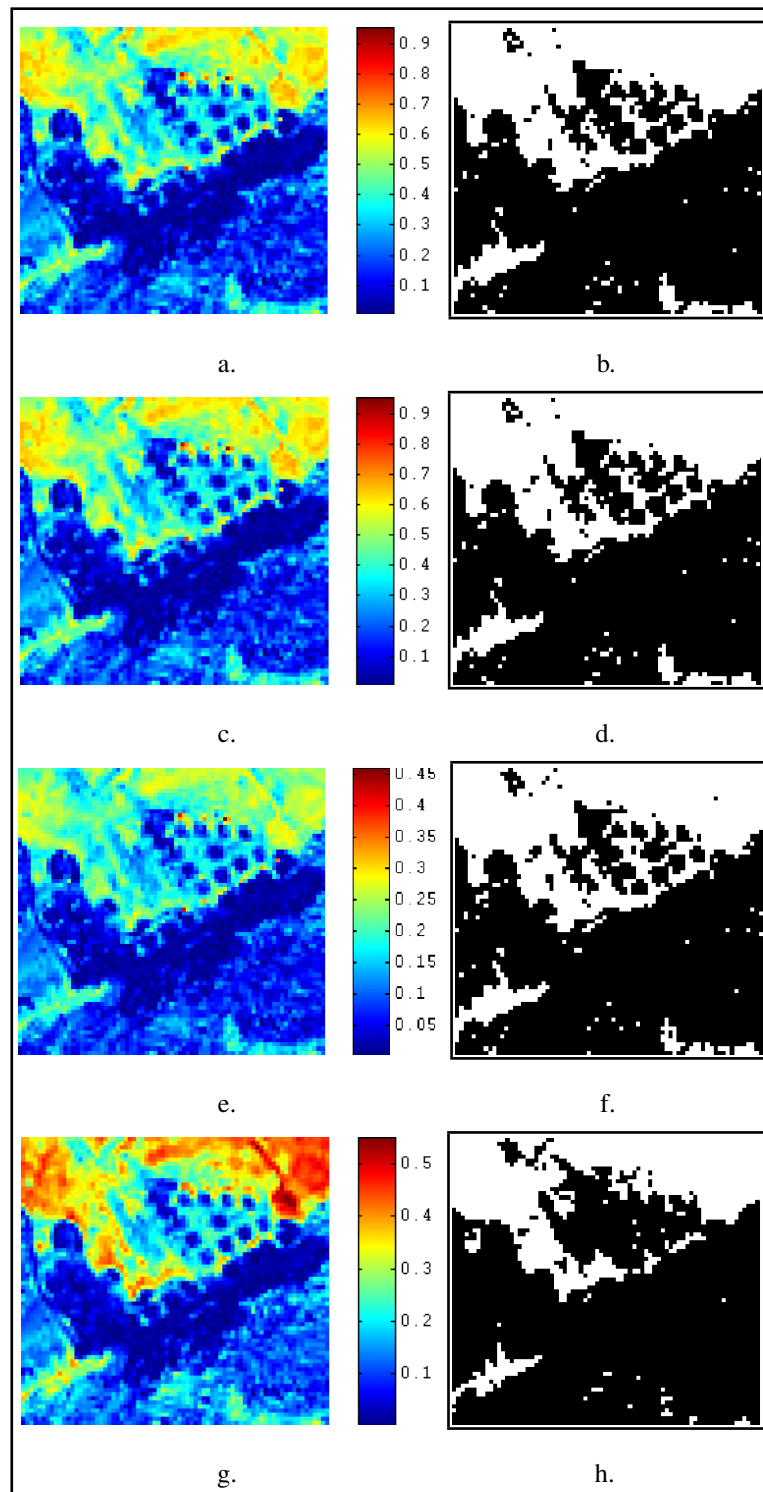


Figure 3.3. Difference of RVI, NDVI, TVI and SAVI for the Adana image set: a. Difference of RVI; c. Difference of NDVI; e. Difference of TVI; g. Difference of SAVI; b.,d.,f. and h. Thresholded versions

where  $\psi$  represents the angle obtained from RVI and  $\theta$  represents angle obtained from NDVI. Time dependent form of the angle vegetation indices is defined as

$$\psi_t = \frac{4}{\pi} \arctan \left( \frac{n^i}{r^j} \right) \quad (3.12)$$

$$\theta_t = \frac{4}{\pi} \arctan \left( \frac{n^i - r^j}{n^i + r^j} \right) \quad (3.13)$$

where  $i$  and  $j$  are the time indices of the near infrared and red bands. For bi-temporal images  $i$  and  $j$  can take values from  $\{1, 2\}$ . Ünsalan tested TDVIs for every combination of  $i$  and  $j$  and compared them to each other and to the results from image differencing and NDVI differencing techniques. Among all techniques, time dependent RVI ( $\psi_t$ ) produced the best result. In this study, we extend the previous method by using all the multispectral band combinations in index calculation.

In Figure 3.4, we provide the following TDVIs for our Adana image set. We used the percentile thresholding, by 97.5%, in finding the threshold value.

$$\psi'_t = \frac{4}{\pi} \arctan \left( \frac{r^2}{g^1} \right) \quad (3.14)$$

$$\psi''_t = \frac{4}{\pi} \arctan \left( \frac{r^2}{b^1} \right) \quad (3.15)$$

$$\theta'_t = \frac{4}{\pi} \arctan \left( \frac{r^2 - g^1}{r^2 + g^1} \right) \quad (3.16)$$

$$\theta''_t = \frac{4}{\pi} \arctan \left( \frac{r^2 - b^1}{r^2 + b^1} \right) \quad (3.17)$$

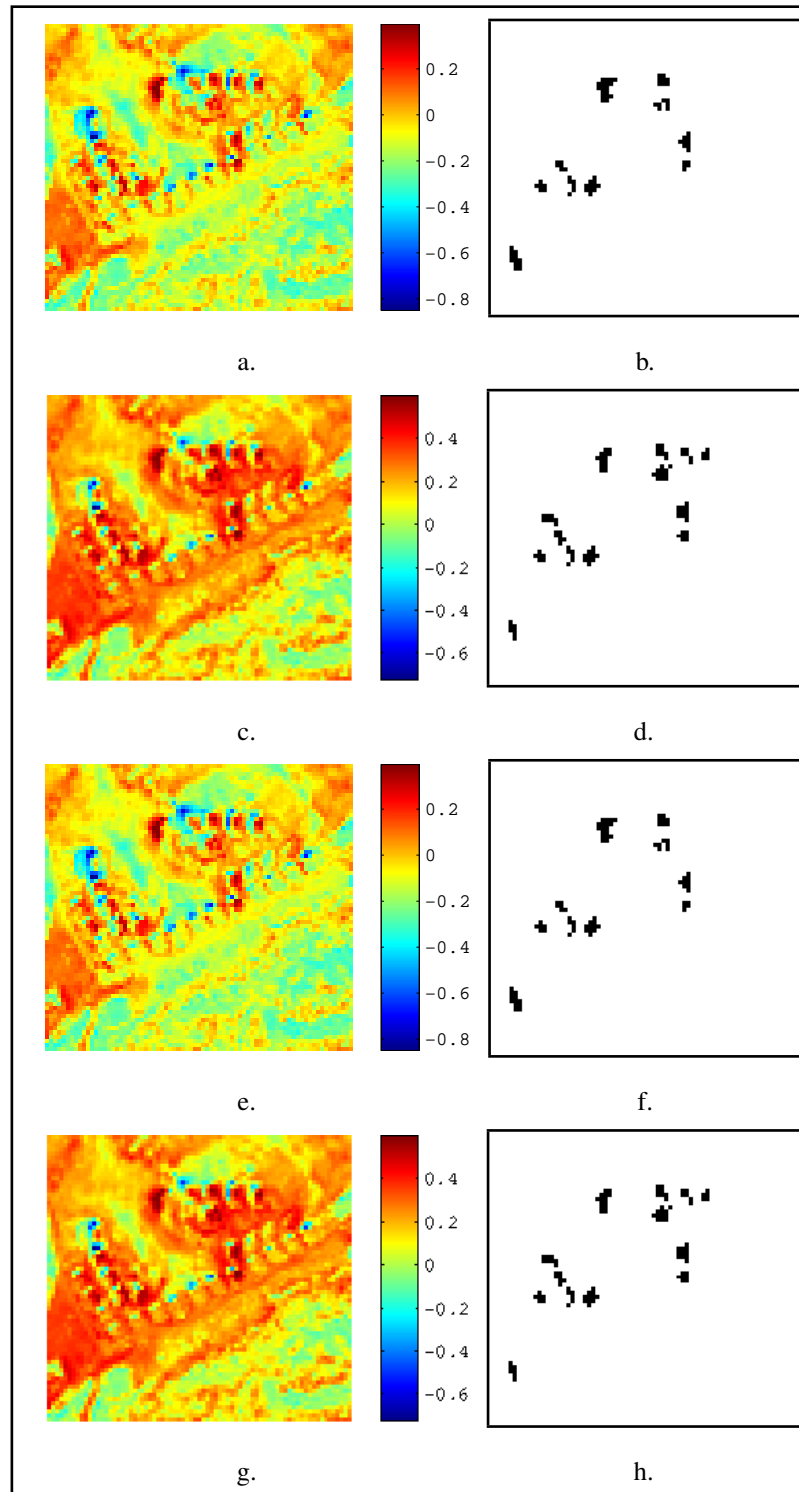


Figure 3.4.  $\psi'_t$ ,  $\psi''_t$ ,  $\theta'_t$  and  $\theta''_t$  applied to the Adana image set: a.  $\psi'_t$ ; c.  $\psi''_t$ ; e.  $\theta'_t$ ; g.  $\theta''_t$ ;  
b.,d.,f. and h. Thresholded versions

### 3.5. COLOR INVARIANTS

Color invariants are transformations of the color images based on the correlations between multiple bands. These correlations are between the sensor response of the camera described in two parts: body reflection and specular reflection [27]. Gevers and Smeulders [28] evaluated the invariance of several transformations from RGB color space in terms of sensor responses based on the following criteria: Viewing direction and object geometry, illumination direction, intensity of the illumination, varying illumination color. They showed that ratio of sum of the sensor responses are insensitive to surface orientation (object geometry), illumination direction and illumination intensity. They proposed the following color invariants based on this information.

$$c_1 = \arctan\left(\frac{r}{\max(g, b)}\right) \quad (3.18)$$

$$c_2 = \arctan\left(\frac{g}{\max(r, b)}\right) \quad (3.19)$$

$$c_3 = \arctan\left(\frac{b}{\max(r, g)}\right) \quad (3.20)$$

Furthermore, they showed that ratio of sum of differences of the sensor responses is insensitive to highlights (e.g. specular reflectance) as well as surface orientation, illumination direction and illumination intensity. They proposed the following color invariants based on this information.

$$l_1 = \frac{(r - g)^2}{(r - g)^2 + (r - b)^2 + (g - b)^2} \quad (3.21)$$

$$l_2 = \frac{(r - b)^2}{(r - g)^2 + (r - b)^2 + (g - b)^2} \quad (3.22)$$

$$l_3 = \frac{(g - b)^2}{(r - g)^2 + (r - b)^2 + (g - b)^2} \quad (3.23)$$



Gevers and Smeulders pointed out to the trade off between the discriminative power and the invariance of the color invariants. Suppose that color model A is invariant to illumination conditions  $w, x, y$  and  $z$ , and color model B is invariant to illumination conditions  $y$  and  $z$  only. Under the illumination conditions where  $w, x, y$  and  $z$  are uncontrolled (varies from sample to sample) color model A produces better results than B. On the other hand, the color model B gives better results than A under the illumination conditions where  $w$  and  $x$  are controlled and  $y$  and  $z$  are uncontrolled. Hence, while the invariance of the color model increases, its discriminative power decreases. For this reason, Gevers and Smeulders proposed color invariants for several invariance levels.

In change detection applications, highlights and illumination color are controlled imaging conditions. Therefore, we need surface orientation and illumination intensity invariant models. Gevers and Smeulders recommended  $(c_1, c_2, c_3)$  color model for this type of invariance. As in the other transformation based change detection methods, we can apply any of the pixel based methods after transformation.

In Figure 3.5, we provide the differences of  $c_1, c_2, c_3$  color invariants for our Adana test image set. As can be seen,  $c_1$  and  $c_3$  color invariants mostly emphasize shadow regions in images. Therefore, they can not be used for change detection directly. On the other hand, difference of the  $c_2$  color invariant emphasizes significant changes such as missing and developing buildings. Hence, it can be used for change detection. In Figure 3.6, we provide its thresholded version. As can be seen, while some of the important changes are kept, several other minor changes are enchanted. For this reason, the performance of  $c_2$  difference is no better than simple image differencing.

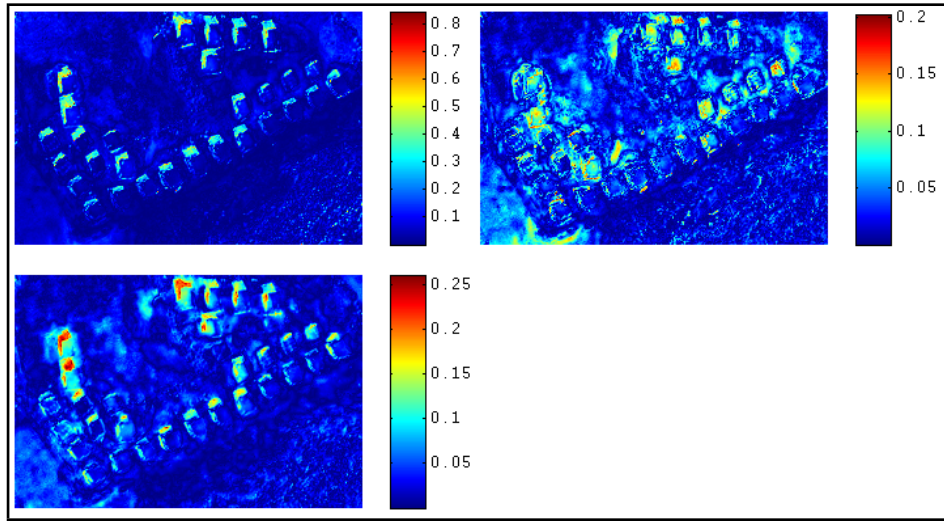


Figure 3.5. Difference of  $c_1$ ,  $c_2$ ,  $c_3$  for the Adana image set



Figure 3.6. Thresholded difference of  $c_2$  for the Adana image set

## 4. TEXTURE ANALYSIS BASED METHODS

In this chapter, we provide two texture based change detection methods. In both methods, we calculate the texture descriptors for bitemporal images separately. In order to detect possible changes, we find their difference. We start with gray level co-occurrence matrix based texture descriptors next.

### 4.1. GRAY LEVEL CO-OCCURRENCE MATRIX

Texture analysis focuses on the statistical explanation to the spatial distribution of the image pixels in a given image. There are several texture analysis methods proposed in the literature. A frequently used one is Gray Level Co-occurrence Matrix (GLCM) introduced by Haralick *et al.* [29]. GLCM entries are number of occurrences of spatial adjacency of gray tone values in an image. Adjacency is defined by the distance in pixel units. An occurrence of spatial adjacency of two gray tones is shown in Figure 4.1. In this sample, gray tones are (41) and (42) and the adjacency is defined as two pixels.

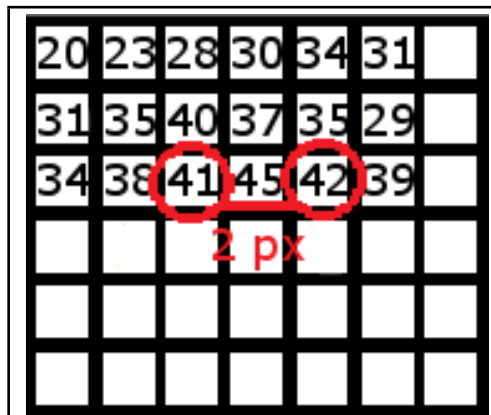


Figure 4.1. Pixels in red circles are adjacent and distance between them is two pixels

In Figure 4.1, adjacent pixels lie on a horizontal line. When we calculate the GLCM by counting only horizontal adjacency values, we find the zero degree ( $0^\circ$ ) co-occurrence matrix. Similarly, we can calculate  $45^\circ$ ,  $90^\circ$  and  $135^\circ$  co-occurrence matrices by counting in these directions. In Figure 4.2,  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  and  $135^\circ$  adjacency values for two pixel distance

are given.

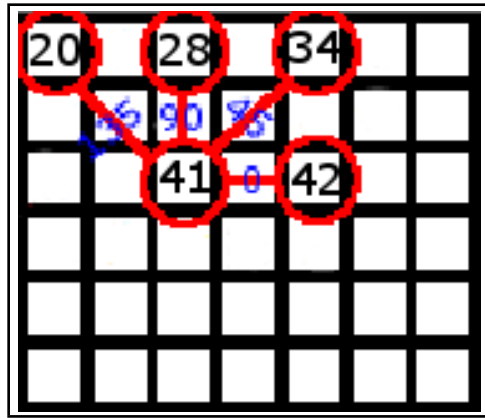


Figure 4.2. Four adjacent pixels of a pixel from four directions

The formal definition of GLCM is as follows. Let  $I(x, y)$  be a grayscale image which takes values from the set  $G = \{1, 2, \dots, N_g\}$ . The horizontal coordinate  $x$  of  $I(x, y)$  takes values from the set  $L_x = \{1, 2, \dots, N_x\}$ . The vertical coordinate  $y$  of  $I(x, y)$  takes values from the set  $L_y = \{1, 2, \dots, N_y\}$ . Then, the image  $I(x, y)$  can be defined as a function from the set  $L_x \times L_y$  to  $G$ .

The relative frequency of the spatial adjacency of gray tones  $i$  and  $j$  is mathematically defined for four directions as

$$P(i, j, d, 0^\circ) = \#\{((k, l), (m, n)) \in (L_y \times L_x) \times (L_y \times L_x) \mid$$

$$k - m = 0, |l - n| = d, I(k, l) = i, I(m, n) = j\} \quad (4.1)$$

$$P(i, j, d, 45^\circ) = \#\{((k, l), (m, n)) \in (L_y \times L_x) \times (L_y \times L_x) \mid$$

$$(k - m = d, l - n = -d) \text{ or } (k - m = -d, l - n = d),$$

$$I(k, l) = i, I(m, n) = j\} \quad (4.2)$$

$$P(i, j, d, 90^\circ) = \#\{(k, l), (m, n) \in (L_y \times L_x) \times (L_y \times L_x) \mid$$

$$|k - m| = 0, l - n = d, I(k, l) = i, I(m, n) = j\} \quad (4.3)$$

$$P(i, j, d, 135^\circ) = \#\{(k, l), (m, n) \in (L_y \times L_x) \times (L_y \times L_x) \mid$$

$$(k - m = d, l - n = d) \text{ or } (k - m = -d, l - n = -d),$$

$$I(k, l) = i, I(m, n) = j\} \quad (4.4)$$

where  $d$  is the adjacency value,  $\#$  sign indicates the number of occurrences under the given conditions. As we mentioned earlier, GLCM entities are number of occurrences. Therefore,  $i$  and  $j$  are matrix indices and GLCM is an  $N_g$  by  $N_g$  matrix.

Haralick *et al.* calculated an average matrix from the four co-occurrence matrices, and used this matrix in calculating texture features. Thus, extracted features are rotation invariant for  $45^\circ$  of rotation. The average matrix is defined as

$$P(i, j, d) = \frac{1}{4} \sum_{n=0}^3 P(i, j, d, n \times 45^\circ) \quad (4.5)$$

Several texture features can be extracted from GLCM. The most useful ones for our purposes

are as

$$\text{Contrast: } \sum_i \sum_j (i - j)^2 P(i, j, d) \quad (4.6)$$

$$\text{Correlation: } \frac{\sum_i \sum_j (ij) P(i, j, d) - \mu_x \mu_y}{\sigma_x \sigma_y} \quad (4.7)$$

$$\text{Energy: } \sum_i \sum_j P(i, j, d)^2 \quad (4.8)$$

$$\text{Inverse Difference Moment (IDM): } \sum_i \sum_j \frac{P(i, j, d)}{1 + (i - j)^2} \quad (4.9)$$

where  $\mu_x$ ,  $\mu_y$ ,  $\sigma_x$  and  $\sigma_y$  are the means and standard deviations of the marginal probability matrices  $P_x = \sum_{j=1}^{N_g} P(i, j, d)$  and  $P_y = \sum_{i=1}^{N_g} P(i, j, d)$ .

As far as change detection is concerned, it is obvious that we can measure the amount of change by finding the difference in the amount of texture. Here, our approach must be slightly different from the pixel based methods since in texture analysis we study a region. One appropriate method is dividing the image into smaller windows and calculating texture features for these. This way, we can calculate differences in texture by comparing the features window by window. As in the pixel based methods, for a healthy comparison, images must be geometrically registered.

Tomowska *et al.* [30] applied texture analysis methods to detect changes in satellite imagery. They used four change detection methods as image differencing, image ratioing, image regression, and PCA. These methods are applied on texture features instead of the intensity values of the image. Transformations are carried out via GLCM. They benefit from four texture features as: contrast, correlation, energy and IDM. Texture features are calculated in different window sizes ranging from  $3 \times 3$  to  $13 \times 13$ . The best results were obtained using  $13 \times 13$  windows. Tomowska *et al.* presented 16 change images from application of four change detection methods with four texture features. Performance of each application is assessed based on the visual output. They reported the use of PCA with

the energy feature produced best results.

We now give results from texture analysis applied to our test images. We calculated contrast, energy, and inverse difference moment features for  $13 \times 13$  windows on our Adana test image set. We applied image differencing, image ratioing and PCA using texture features. All change detection methods produced similar results. Therefore, we only present results from image differencing in Figure 4.3 where the difference images calculated using texture features are presented. In obtaining the change map, we apply percentile thresholding, percentile being 97.5%.

## 4.2. ENTROPY

Entropy is a measure of randomness and can be used in quantifying texture [31]. The formal definition of entropy is

$$E = \sum_{i=0}^{N_g-1} p_i \log_2 p_i \quad (4.10)$$

where  $p_i$  is the relative frequency of the intensity levels in the region and  $N_g$  is the number of possible intensity levels.

In using the entropy for change detection, we apply the following strategy. We calculate the entropy of our test images separately for  $11 \times 11$  windows around each pixel. For the Adana test image set, we provide the obtained results in Figure 4.4. As can be seen, developing regions are well emphasized by the second entropy image.

In our tests, we observed that taking the difference of entropy images does not produce expected results. Therefore, we first thresholded the images separately, then obtain their difference. We provide the thresholded (by 80% percentile) entropy images in Figure 4.5.a and Figure 4.5.b. Then, we obtain the difference by the binary xor operation. To refine results, we removed noise in the difference image by applying morphological opening with a disk shaped structuring element. We provide the final result in Figure 4.5.c.

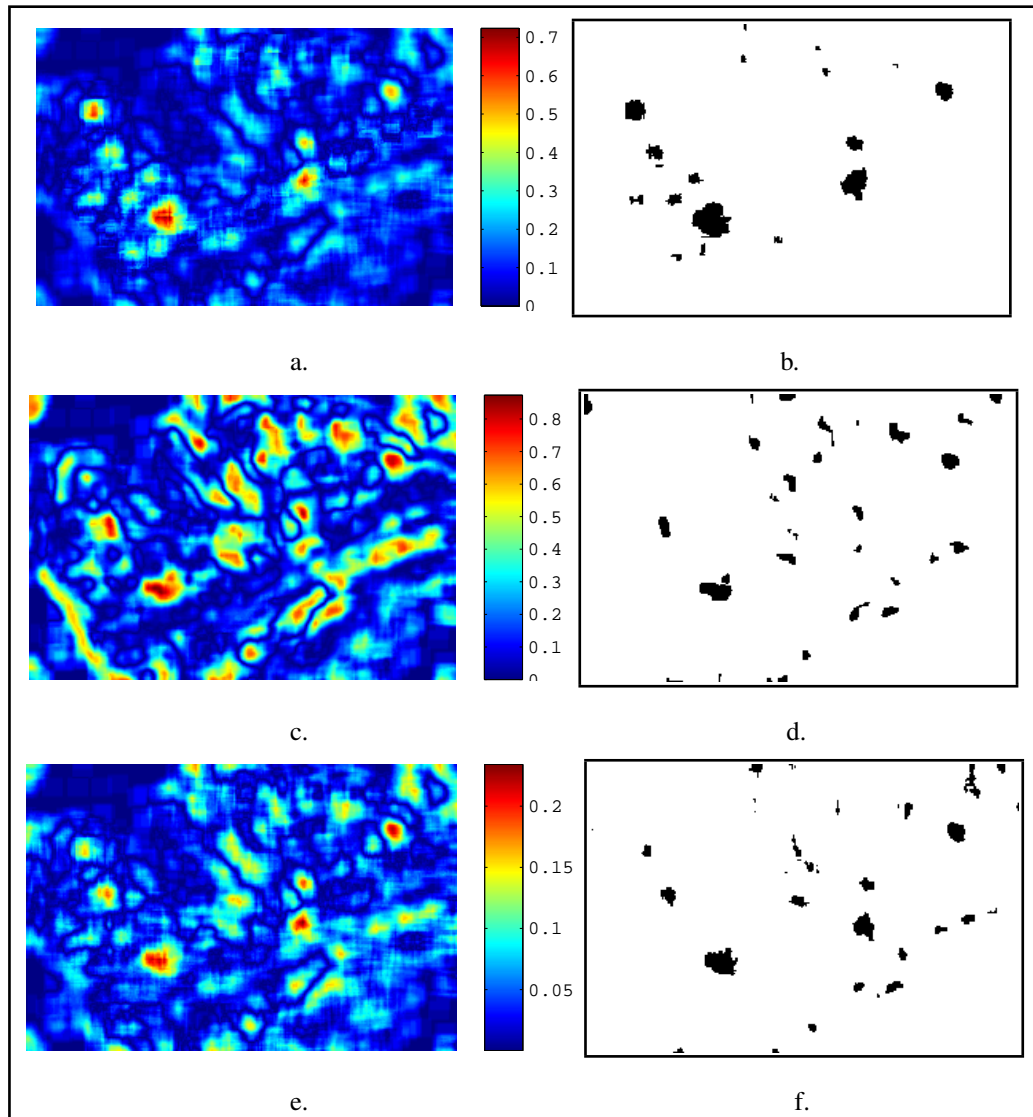


Figure 4.3. Image differencing applied on GLCM features: a. Difference of the contrast; c. Difference of the energy; e. Difference of the IDM; b.,d. and f. Thresholded versions

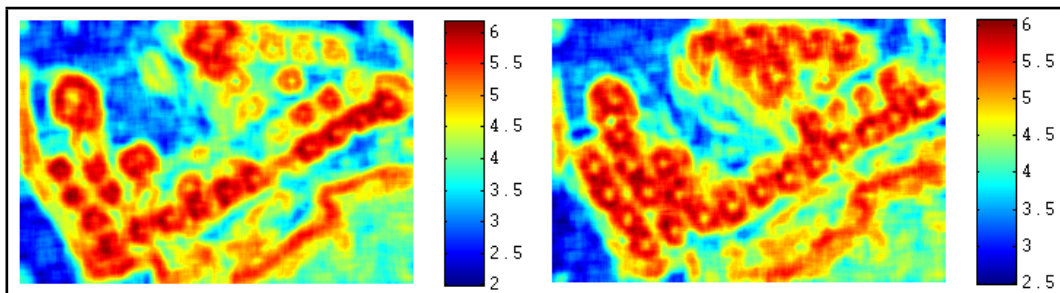


Figure 4.4. Entropy images calculated for test images



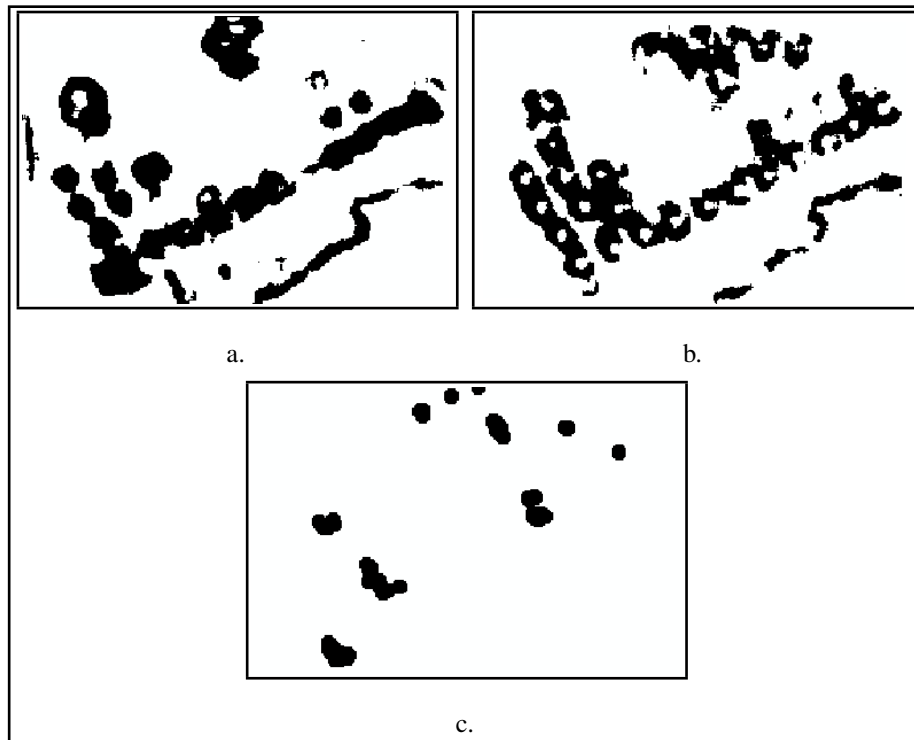


Figure 4.5. Change detection by entropy texture feature: a. and b. Thresholded entropy images; c. The final difference image obtained by entropy texture feature

## 5. STRUCTURE BASED METHODS

This chapter deals with change detection methods based on the structure information in bitemporal images. We define the structure in various ways. In the following sections, we explore each method in detail.

### 5.1. EDGE DETECTION

The first method for structural change detection is based on edge information. We obtain the edge pixels from two images using Canny's edge detection method [32]. For the registered image pairs, we expect to have correspondence between edge maps of the two images. However, direct comparison is not feasible. Although the images can be registered, their looking angle may not be the same. Moreover, there may be shadow effects. Therefore, instead of finding the difference of both edge maps, we obtain the connected components between them [33]. If some parts of the edge map are not matched, then that region is taken as change.

In Figure 5.1, we provide the edge maps obtained from the Adana image set. In Figure 5.2, we provide the non-matched edge pixels. These represent the changed areas in the Adana image set.

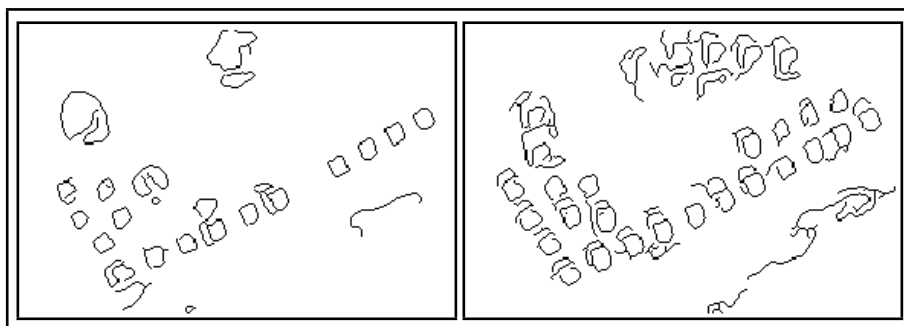


Figure 5.1. Edge maps obtained from the Adana image set

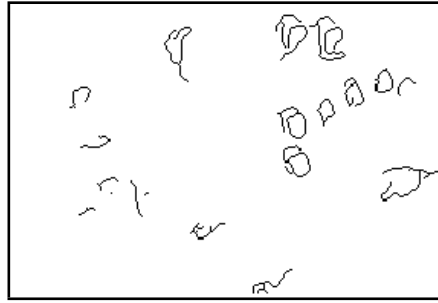


Figure 5.2. Change detection results from the edge map matching of the Adana image set

## 5.2. GRADIENT MAGNITUDE BASED SUPPORT REGIONS (GMSR)

Similar to edge based change detection, we can represent the edge information by gradient magnitude based support regions. The GMSR is introduced in a previous study for land classification [34]. Here, we apply the same methodology as we had done for edge detection with GMSR at hand. We provide the GMSR obtained for both images in the Adana image set in Figure 5.3. We provide the change detection results in Figure 5.4.

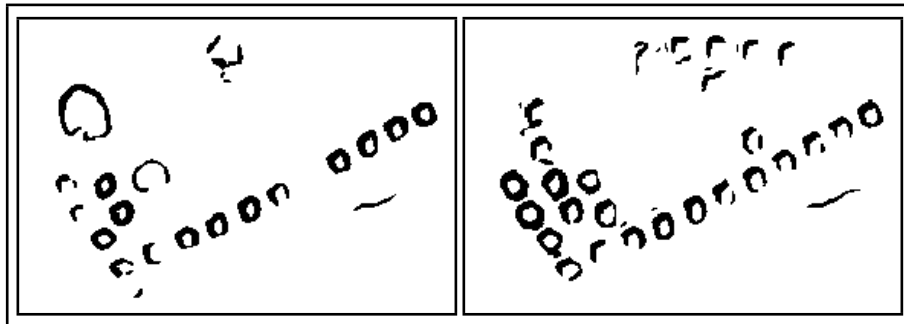


Figure 5.3. GMSR obtained from the Adana image set

## 5.3. MATCHED FILTERING

In the matched filtering approach, we assume a generic shape for buildings in the image. Using the prototype for this shape, we detect buildings in images using the matched filtering approach. Matched filtering is a standard method for digital communication systems [35]. In this thesis, we picked the Laplacian of Gaussian (LoG) filter as a generic building shape. We apply it to both images and obtain high response regions, possibly representing buildings.

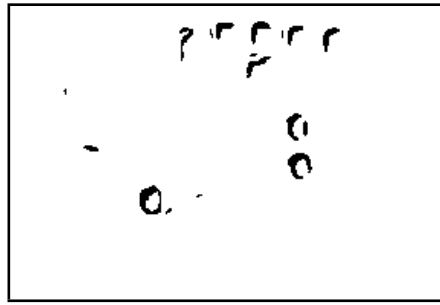


Figure 5.4. Change detection results from the GMSR matching of the Adana image set

After thresholding, we apply the same methodology as we had done in edge detection approach. We provide the thresholded LoG responses for both Adana images in Figure 5.5. We provide the change regions detected by this method in Figure 5.6.

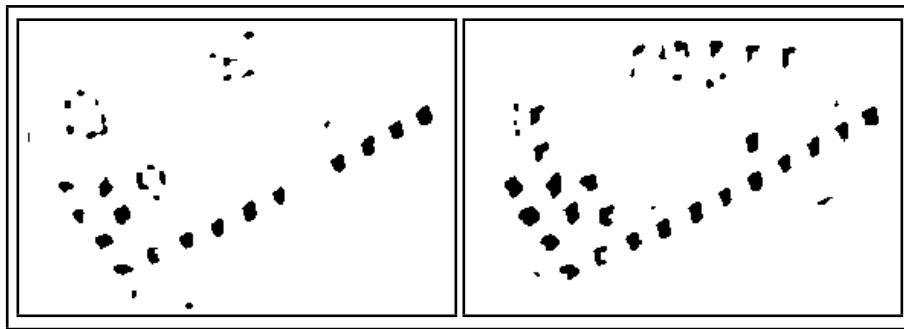


Figure 5.5. GMSR obtained from the Adana image set

#### 5.4. MEAN SHIFT SEGMENTATION

Mean shift segmentation is introduced by Comanicu and Meer [36] as an application of feature space analysis discussed in the same study. They referred to the density estimators for cluster analysis and in particular to the kernel density estimators. They showed that, mean shift vectors (obtained after the calculation of the density gradient) can be used for finding the local maxima points in the feature space. Their feature space formation consists of both spatial domain information and spectral domain information. For segmentation, they first applied a mean shift based edge preserving smoothing filter to the images. Then, they found the segments by delineating the feature space clusters which are the groups of mean shift filtered image pixels that belong to the same basin of attraction. A basin of attraction



Figure 5.6. Change detection results from the GMSR matching of the Adana image set

is defined as the set of all locations that converge to the same mode.

As far as change detection is concerned, similar to edge based information, we consider segmentation of both images and detect changes based on segments. We refine segments based on their shape information using two region based shape descriptors as area and solidity [33]. Area is the number of pixels the region contains. Solidity is the ratio between the area and the area of the convex hull of the region. We eliminate segments having area greater than a threshold value and solidity less than a threshold value. Finally, we apply the methodology which we defined for the edge based comparison to the segments to detect changes. We provide the segments obtained by mean shift clustering and shape refinement in Figure 5.7. We provide the changed regions in Figure 5.8.

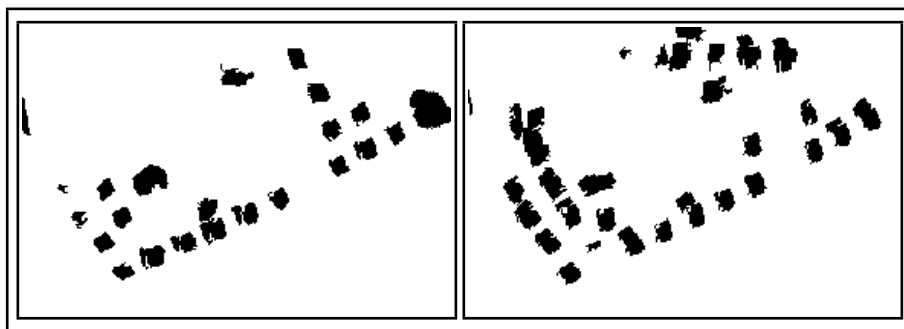


Figure 5.7. Segments obtained from the Adana image set

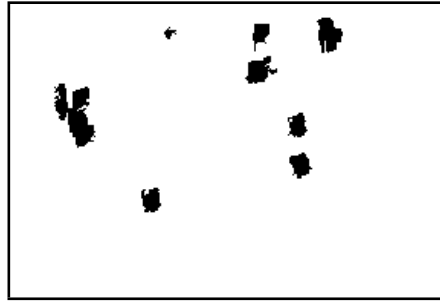


Figure 5.8. Change detection results from the segments of the Adana image set

### 5.5. LOCAL FEATURES

We can describe any object in an image if we can find some distinctive features for it. Once extracted, these features can be used for recognizing the object in another image by comparing either their spatial locations or their descriptors. Local feature detectors try to isolate these features. Lowe [37] pointed out the distinctiveness of the features as “The features must also be sufficiently distinctive to identify specific objects among many alternatives. The difficulty of the object recognition problem is due in large part to the lack of success in finding such image features. However, recent research on the use of dense local features has shown that efficient recognition can often be achieved by using local image descriptors sampled at a large number of repeatable locations”. He also refers to the repeatability of the local features in this quotation. A feature detector should also be able to find distinctive features at the same location under varying conditions such as scaling or illumination.

In this section, we propose a new change detection method based on local feature matching. We use two widely used local feature detectors: Scale Invariant Feature Transform (SIFT) and Features from Accelerated Segment Test (FAST) [37, 38]. SIFT is a local feature detector with several valuable properties such as invariance to image scaling, translation, and rotation, and partial invariance to illumination changes and affine or 3D projection. FAST is a high-speed feature detector. It targets real-time applications which are short in processing time.

Our method proceeds in two steps: matching keypoints(local features) and finding changed regions. Keypoint matching is applied in a similar manner for SIFT and FAST keypoints. FAST keypoints are matched as follows. Keypoints in the first image are assumed to be matched to a keypoint in the second image if they are spatially close. Closeness norm is determined by fixed threshold values such as  $dx$  and  $dy$  for two dimensions. In this way, if a keypoint in the first image is located at  $(x, y)$ , we search the existence of a keypoint in a region bounded by  $[x - dx \ x + dx]$  and  $[y - dy \ y + dy]$  in the second image. For SIFT keypoint matching, in addition to the spatial constraint, we also compare the feature descriptor distances. Descriptor of the keypoint in the first image is compared to the descriptor of the keypoints from the previously defined bounded region. If there is a sufficiently close keypoint, then it is assumed to be matched. No-matched keypoints are used for finding the change mask. Centered at each non-matched keypoint, a Gaussian with a fixed variance is added on to the mask. Gaussians are added up to determine the change density distribution. Finally, a threshold is applied to extract the change region.

We provide the change detection results for the Adana image set for three cases: matching only SIFT keypoints, matching only FAST keypoints and sum of separate results (SIFT and FAST). We provide the change detection results obtained by matching SIFT keypoints in Figure 5.9. In Figure 5.9.a and Figure 5.9.b sum of the Gaussians centered at each non-matched keypoint is presented for test images. Separate change density distributions are added up resulting in total change as provided in Figure 5.9.c. The change density function is thresholded by 90%4 percentile, and the change regions are extracted. The result of this operation is provided in Figure 5.9. In a similar manner, we provide the change detection results obtained by matching FAST keypoints in Figure 5.10. Finally, we provide the change detection results obtained after combining the SIFT and FAST results in Figure 5.11.

## 5.6. BIPARTITE GRAPH MATCHING

In this section, we propose a novel change detection algorithm based on the graph based representation of the structural information. Our focus is detecting changes in a specific region using local features in a graph formalism. To represent the structure, we extract local features from both images using FAST. Then, we represent each local feature set (extracted

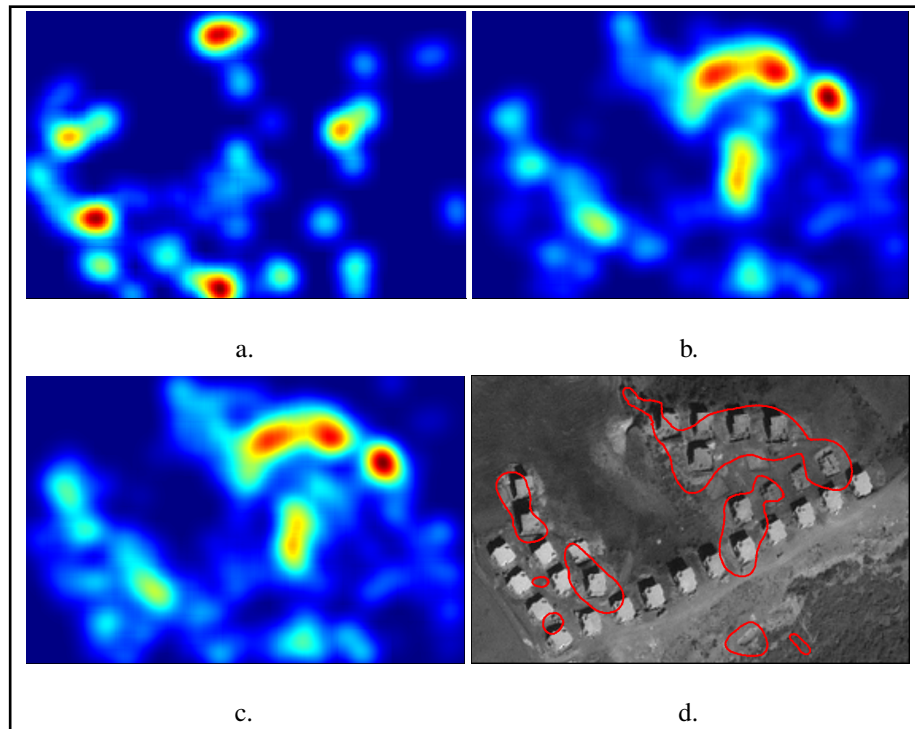


Figure 5.9. Change detection using SIFT features: a. and b. Change density distribution for Adana images; c. Total change density distribution; d. Thresholded version



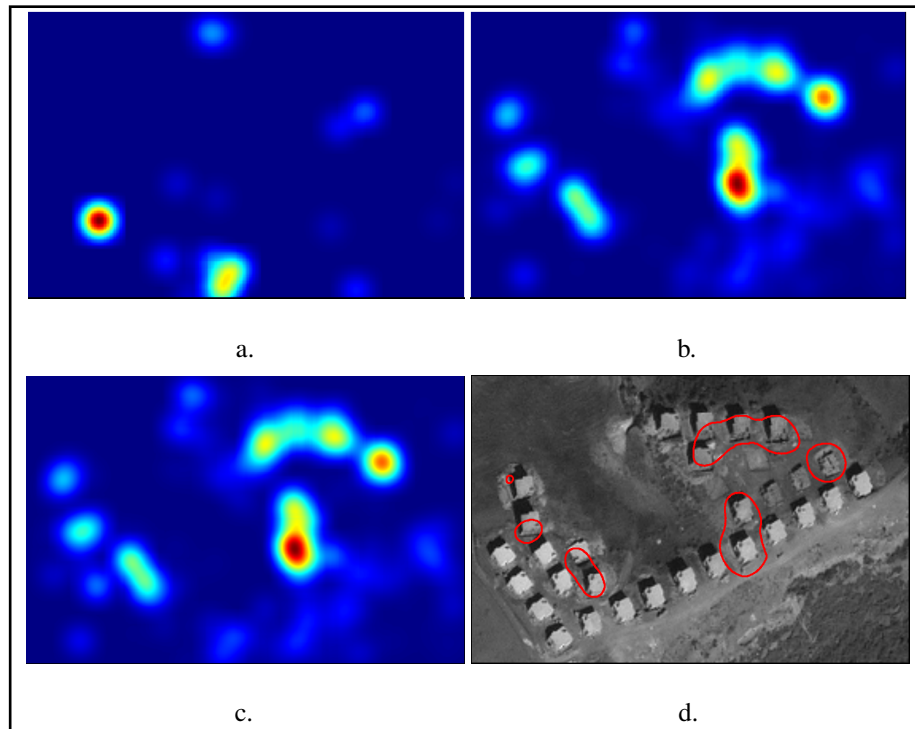


Figure 5.10. Change detection using FAST features: a. and b. Change density distribution for Adana images; c. Total change density distribution; d. Thresholded version

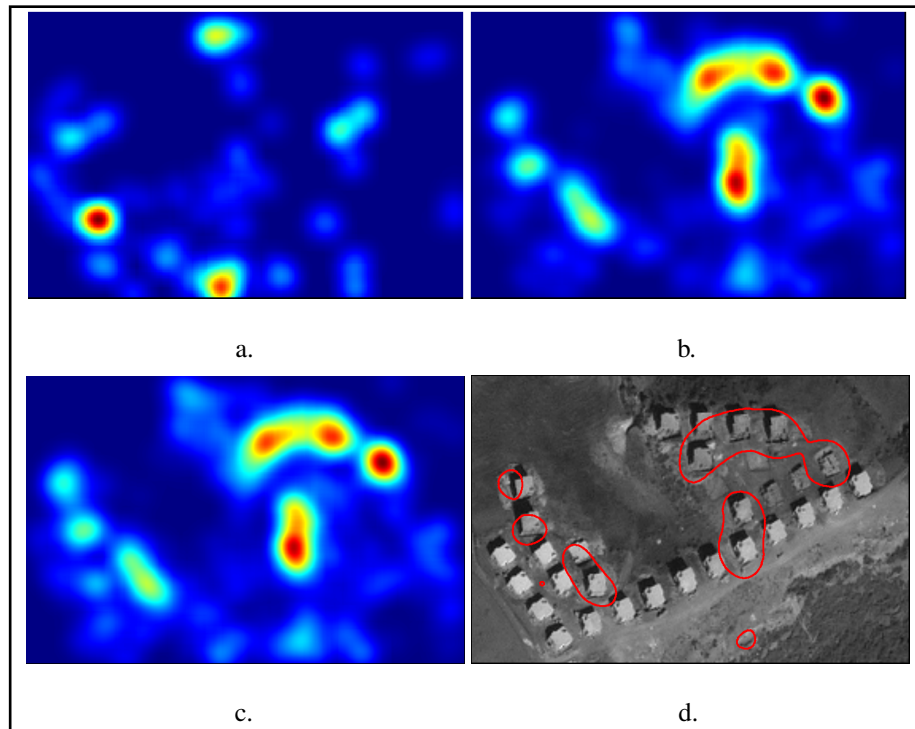


Figure 5.11. Change detection using the combination of FAST and SIFT features: a. and b. Change density distribution for Adana images; c. Total change density distribution; d. Thresholded version

from different images) in a graph formation separately. This allows us to detect changes using a graph matching method.

To extract the structure information from local features, we represent them in a graph form. A graph  $G$  is represented as  $G = (V, E)$ , where  $V$  is the vertex set and  $E$  is the edge matrix showing the relations between these vertices. Here vertices are local features extracted by FAST. The edges are formed between them just by their distance. If a distance between two vertices are small, there will be an edge between them. In this study, we set this difference value to 10 pixels depending on the characteristics of the objects in the image.

As we form graphs from both images separately, we apply graph matching between them. In matching graphs, we apply constraints both in spatial domain and in neighborhood. We can summarize this method as follows. Let the graph formed from the first and second images be represented as  $G_1(V_1, E_1)$  and  $G_2(V_2, E_2)$ . In these representations,  $V_1 = \{f_1, \dots, f_n\}$  holds the local features from the first image and  $V_2 = \{s_1, \dots, s_m\}$  holds the local features from the second image. We first take spatial constraints in graph matching. We assume that two vertices match if the spatial distance between them is smaller than a threshold. In other saying,  $f_i$  and  $g_j$  are said to be matched if  $\|f_i - g_j\| < \delta$ ,  $\delta$  being a threshold. This threshold adds a tolerance to possible image registration errors. Non-matched vertices from both graphs represent possible changed objects (represented by their local features). Since local features simply represent a single location, we add a circular tolerance to the non-matched ones to represent possible change area. We can also add neighborhood information to graph matching. To do so, we first eliminate vertices having neighbors less than a number. Then, we match these refined vertices. This way, we eliminate some local features having no neighbors (possible noise regions).

We provide the local features extracted from the Adana image set in Figure 5.12. We provide the change detection results with just using spatial constraints and three neighborhood constraints in Figure 5.13.

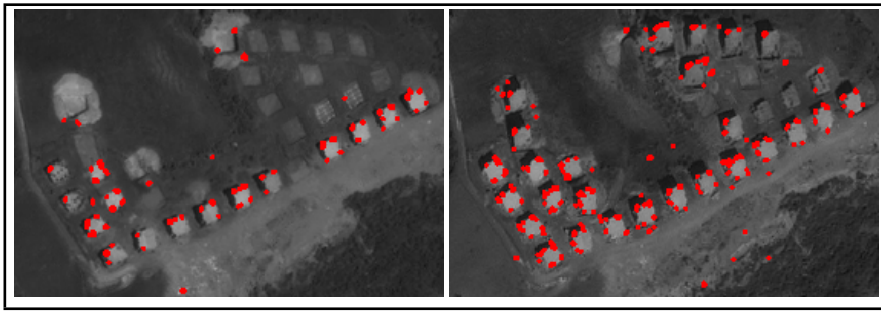


Figure 5.12. Local features extracted by FAST which are used in graph formation.



Figure 5.13. Graph matching based comparison results: a. Using only spatial constraints; b. Adding three neighborhood constraint

## 5.7. SHADOW INFORMATION

Shadow in a region gives an indirect information about the elevation of the objects. This information can not be extracted by the techniques (that we considered so far in this study). Therefore, it is valuable. Elevation information can be helpful especially in urban monitoring, since manmade structures have height. In this study, we will use the shadow information for two dimensional change detection. We benefit from Sırmaçek and Ünsalan's study. They used

$$\psi_b = \frac{4}{\pi} \arctan \left( \frac{b - g}{b + g} \right) \quad (5.1)$$

for shadow detection. In this equation,  $b$  and  $g$  are the blue and green bands of the image.

In detecting changes, we compare the shadow information extracted from times  $t1$  and  $t2$ . Based on these, we estimate the spatial location of the change. In this method, images must be registered (as in most other change detection techniques) before comparison. Additionally, images must be taken at the same time of the year for a healthy comparison. As a final comment, the detected change locations are in terms of shadows. Therefore, they only represent the changed objects' shadows.

Our method has three steps. First, we extract shadow indicators by applying  $\psi_b$  to the test images. Then, we threshold the results and obtain the shadow regions. Finally, we detect shadow regions that does not exist in the other image. We provide the shadow indicators for our test images in Figure 5.14. The thresholded (using Kapur's method) results are given in Figure 5.15. Finally, we eliminate shadow regions that overlap with any shadow region in the other image in pixel basis. Remaining regions are accepted as shadow difference. We provide the obtained results in Figure 5.16. While some new shadow regions are missed, most of the regions are detected. We note that detected regions are located around the developing parts of the area.

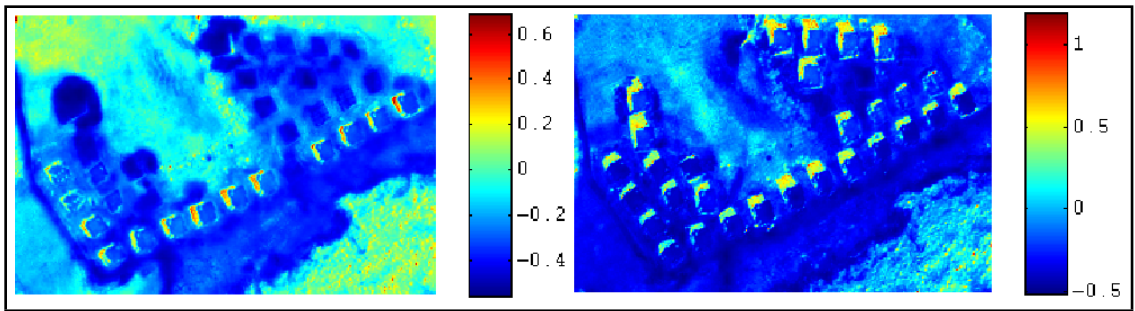


Figure 5.14.  $\psi_b$  applied to test images



Figure 5.15.  $\psi_b$  results are threshold by Kapur's algorithm



Figure 5.16. Shadow differences plotted on the second image

## 6. EXPERIMENTS

In this chapter, we provide experimental results obtained from all change detection methods considered in this study. We first explain the data set used in the experiments. Then, we provide performance results in tabular form for each change detection method group in detail.

### 6.1. THE DATA SET

Our data set consists of images which are extracted from high resolution Ikonos images. These are acquired from the particular regions of Ankara and Adana in four different times between years 2001 and 2004. Besides using three band panchromatic images, we also used their four band multi-spectral versions. Panchromatic Ikonos images have one meter resolution. Corresponding multi-spectral images have four meter resolution and contain red, green, blue, and near infra red spectral bands. In our data set, 18 image pairs are panchromatic (labeled with letter P in suffix) and 17 image pairs are multi-spectral (labeled with letter M in suffix). Our test images are geometrically registered. Therefore, we did not need to register them. These images are also radiometrically normalized as discussed in Section 2.1. We provide a sample set of our test images in Figures 2.2 and 6.1.

### 6.2. PERFORMANCE TESTS

In evaluating each method, we have a manually generated ground truth image set. In forming the ground truth, we specifically focused on urban changes. Therefore, we mainly labeled the changed regions which occurred in the man-made structures such as buildings and road segments. In Figure 6.2 we provide three ground truth images generated for *AdanaP<sub>2</sub>*, *AnkaraP<sub>3</sub>* and *AnkaraP<sub>10</sub>* image pairs.

In Tables 6.1 and 6.2, we provide number of ground truth pixels (GT) and number of Image Pixels (IP) for each image. ‘GT Pixels/Image Pixels’ ratio (GT/IP) indicates the amount of change involved in the specific image pair. We provide the total panchromatic and multi-

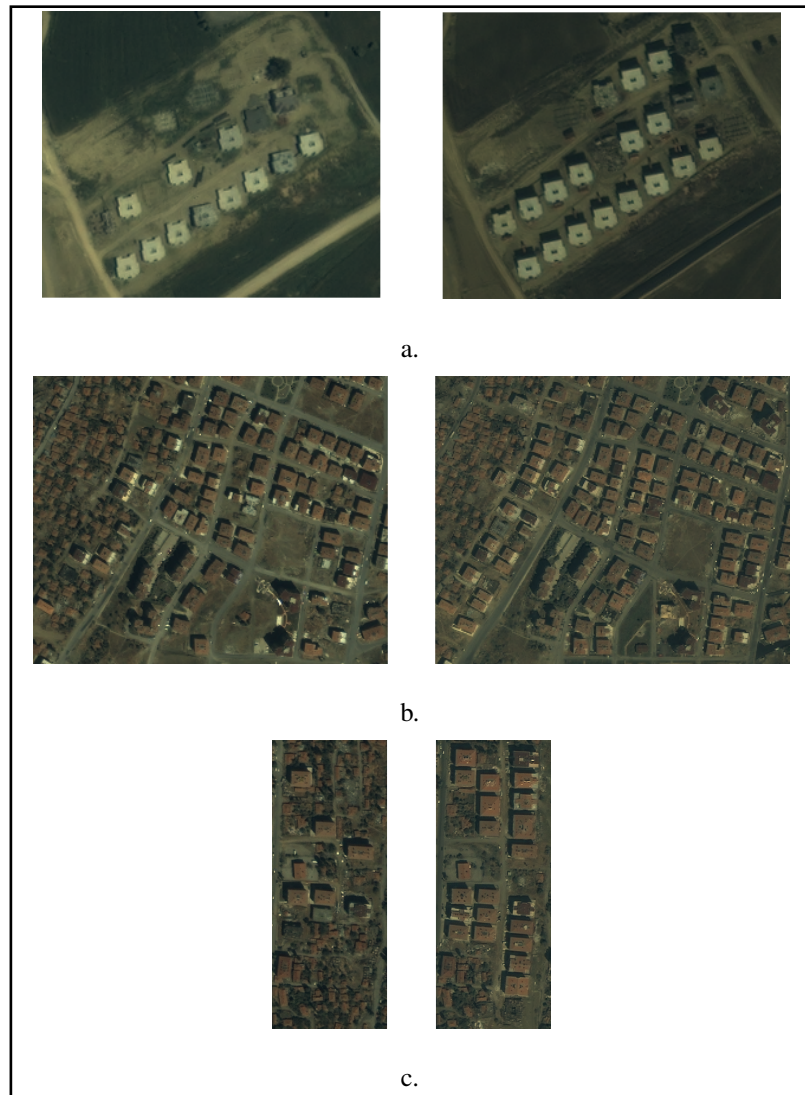


Figure 6.1. Three sample image pairs acquired from Adana and Ankara: a.  $AdanaP_2$ ; b.  $AnkaraP_3$ ; c.  $AnkaraP_{10}$



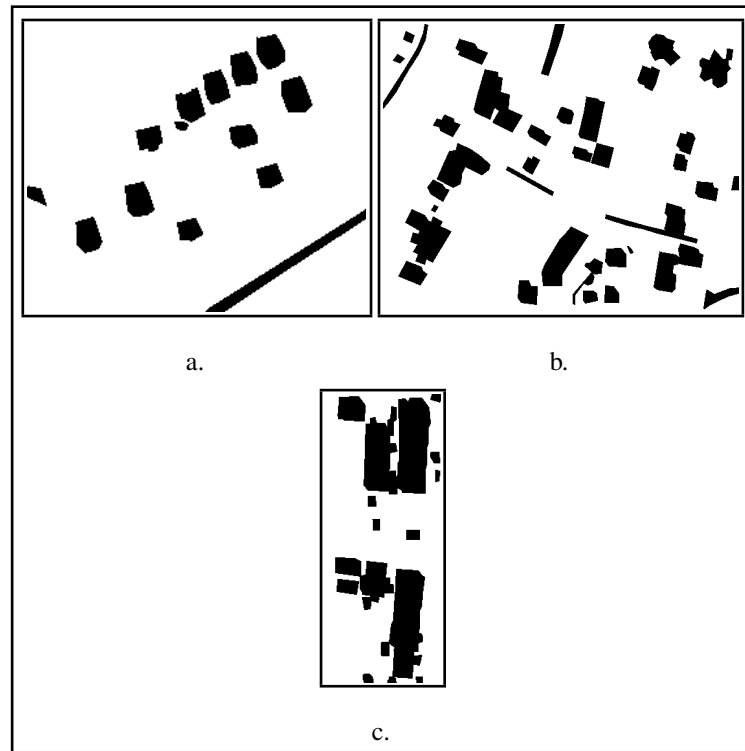


Figure 6.2. Ground truth images for three test image pairs: a. *AdanaP*<sub>2</sub>; b. *AnkaraP*<sub>3</sub>; c. *AnkaraP*<sub>10</sub>

spectral image pixels in Table 6.3.

Table 6.1. Ground truth pixel information for the panchromatic test image set.

	Image Size	GT	IP	GT/IP
<i>AdanaP</i> <sub>1</sub>	270 × 180	5893	48600	0.1213
<i>AdanaP</i> <sub>2</sub>	260 × 220	5500	57200	0.0962
<i>AdanaP</i> <sub>3</sub>	200 × 240	2087	48000	0.0435
<i>AdanaP</i> <sub>4</sub>	200 × 200	7180	40000	0.1795
<i>AdanaP</i> <sub>5</sub>	260 × 400	8383	104000	0.0806
<i>AdanaP</i> <sub>6</sub>	250 × 280	11185	70000	0.1598
<i>AdanaP</i> <sub>7</sub>	350 × 350	18296	122500	0.1494
<i>AdanaP</i> <sub>8</sub>	290 × 440	9549	127600	0.0748
<i>AnkaraP</i> <sub>1</sub>	700 × 700	97684	490000	0.1994
<i>AnkaraP</i> <sub>2</sub>	468 × 477	41573	223236	0.1862
<i>AnkaraP</i> <sub>3</sub>	550 × 445	40931	244750	0.1672
<i>AnkaraP</i> <sub>4</sub>	780 × 350	36525	273000	0.1338
<i>AnkaraP</i> <sub>5</sub>	850 × 760	187970	646000	0.2910
<i>AnkaraP</i> <sub>6</sub>	490 × 520	89876	254800	0.3527
<i>AnkaraP</i> <sub>7</sub>	340 × 330	36074	112200	0.3215
<i>AnkaraP</i> <sub>8</sub>	250 × 310	15800	77500	0.2039
<i>AnkaraP</i> <sub>9</sub>	360 × 340	18878	122400	0.1542
<i>AnkaraP</i> <sub>10</sub>	160 × 400	21215	64000	0.3315

Our comparisons are based on the intersections, unions and complements of the ground truth and the result set. We use four definitions as True Positive ( $TP$ ), False Positive ( $FP$ ), True Negative ( $TN$ ), and False Negative ( $FN$ ). In terms of ground truth set ( $GT$ ) and the result

Table 6.2. Ground truth pixel information for the multi-spectral test image set.

	Image Size	GT	IP	GT/IP
<i>AdanaM</i> <sub>1</sub>	65 × 60	426	3900	0.1092
<i>AdanaM</i> <sub>2</sub>	120 × 140	1386	16800	0.0825
<i>AdanaM</i> <sub>3</sub>	140 × 260	13993	36400	0.3844
<i>AdanaM</i> <sub>4</sub>	75 × 70	381	5250	0.0726
<i>AdanaM</i> <sub>5</sub>	100 × 90	2063	9000	0.2292
<i>AdanaM</i> <sub>6</sub>	90 × 90	1019	8100	0.1258
<i>AdanaM</i> <sub>7</sub>	120 × 100	1571	12000	0.1309
<i>AdanaM</i> <sub>8</sub>	90 × 70	1316	6300	0.2089
<i>AnkaraM</i> <sub>1</sub>	290 × 210	4724	60900	0.0776
<i>AnkaraM</i> <sub>2</sub>	150 × 250	6454	37500	0.1721
<i>AnkaraM</i> <sub>3</sub>	80 × 155	4276	12400	0.3448
<i>AnkaraM</i> <sub>4</sub>	80 × 70	1296	5600	0.2314
<i>AnkaraM</i> <sub>5</sub>	90 × 75	2279	6750	0.3376
<i>AnkaraM</i> <sub>6</sub>	65 × 100	1610	6500	0.2477
<i>AnkaraM</i> <sub>7</sub>	100 × 80	1084	8000	0.1355
<i>AnkaraM</i> <sub>8</sub>	90 × 90	1351	8100	0.1668
<i>AnkaraM</i> <sub>9</sub>	100 × 100	1268	10000	0.1268

Table 6.3. Ground truth total pixel information for panchromatic images and multi-spectral images.

Image Type	Total GT	Total IP	Total GT/IP
Panchromatic	654599	3125786	0.2094
Multi-spectral	46497	253500	0.1834

set ( $RE$ ) these quantities are defined as

$$TP = GT \cap RE \quad (6.1)$$

$$FP = RE - GT \quad (6.2)$$

$$TN = (GT \cup RE)^c \quad (6.3)$$

$$FN = GT - RE \quad (6.4)$$

where ‘-’ sign is the set-theoretic difference and superscript ‘c’ is the set complement. In Figure 6.3, we provide these four set regions.

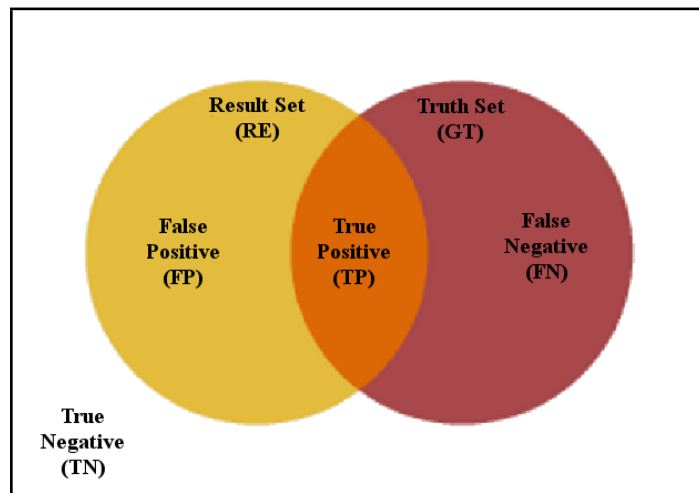


Figure 6.3. Pixel correspondence between the ground truth set and the result set

In terms of pixel based comparison, TP represent the number of change pixels correctly detected; FP represent the number of no-change pixels incorrectly labeled as change by the method; TN represent the number of no-change pixels correct detected; and FN represent the number of change pixels incorrectly labeled as no-change by the method. Based on these quantities, we benefit from three performance criteria as: The percentage correct classification ( $PCC$ ), the Jaccard coefficient ( $Jaccard$ ), and the Yule coefficient ( $Yule$ )

[4, 39]. These are defined as

$$PCC = \frac{TP + TN}{TP + FP + TN + FN} \quad (6.5)$$

$$Jaccard = \frac{TP}{TP + FP + FN} \quad (6.6)$$

$$Yule = \frac{TP}{TP + FP} + \frac{TN}{TN + FP} - 1 \quad (6.7)$$

*PCC* is the most common performance measure. Literally it is the comparison of the truly found pixels to the whole pixel set. However, this measure is not sufficient for our comparisons for images containing little change. From Tables 6.1 and 6.2, we can see that ‘GT/IP’ value can drop as low as 0.043 which indicates too little change. In such cases, any method can get very high *PCC* rating by just labeling all pixels as negative. TN dominates the *PCC* measure and ratings can easily reach one. In order to avoid this situation, we use the *Jaccard* measure. This measure excludes TN in its formulation. *Yule* coefficient gives a measure of association between the result set and the ground truth. *PCC* and *Jaccard* measures are in the range [0 1]. The *Yule* measure is in the range [-1 1]. The higher these measures, the better the performance.

### 6.2.1. Pixel Based Change Detection Methods

We start giving our test results with pixel based methods. Pixel based methods use only panchromatic image pairs. Therefore, in these tests we used only 18 image pairs from our database. Instead of giving results for each image separately, we provide the total image statistics. To this end, we sum TN, TP, FP, and FN for each image and then calculate the performance measures. In Table 6.4 performance measures are provided for pixel based methods. Results are obtained using Otsu’s algorithm.

As can be seen in Table 6.4, image differencing, image regression, and CVA methods produced similar results. Image ratioing produced better *Jaccard* rating and worse *Yule* rating. This indicates that, while it finds more TP pixels than other methods, it labels

Table 6.4. Performance measures for pixel based methods using Otsu's method.

	TP	TN	FP	FN	PCC	Jaccard	Yule
Image differencing	139599	2386573	84614	515000	0.8082	0.1888	0.4451
Image ratioing	171727	2345767	125420	482872	0.8054	0.2202	0.4072
Image regression	141249	2366909	104278	513350	0.8024	0.1861	0.3971
CVA	132453	2393494	77693	522146	0.8081	0.1809	0.4512
Fuzzy XOR	72034	2435920	35267	582565	0.8023	0.1044	0.4783

noise pixels as well. Fuzzy XOR method produced similar *PCC* rating compared to other methods, but it failed to find comparable TP pixels using Otsu's method.

### 6.2.2. Transformation Based Change Detection Methods

We provide performance test results for transformation based methods in Table 6.5. Results are obtained using Otsu's method. First four rows of the table are principal component results. Since we used four band multispectral images in PCA, we found four principal components. We used separate rotation type PCA (Section 3.1). First PC produced results close to pixel based methods. Other principal components produced poor results, even the second PC produced negative *Yule* rating. Low *PCC* and *Yule* ratings are due to the noisy output of these principal components. For the KTT method, the brightness band performed best. Other bands are not so bad, but not close to pixel based methods in terms of the *Jaccard* measure. Vegetation indices RVI, NDVI, TVI, and SAVI performed fairly well. For time-dependent vegetation indices (TDVI), we tested four of them. In their formulation  $r_2$  is the red band from  $t_2$ , and  $g_1$  and  $b_1$  are green and blue bands from  $t_1$ . TDVI2 and TDV4 which use red and blue bands produced better results. High *PCC* and *Yule* ratings indicate that threshold values can be decreased for higher *Jaccard* rating. Color invariant method produced poor results.

Table 6.5. Performance measures for transformation based methods using Otsu's method.

	TP	TN	FP	FN	PCC	Jaccard	Yule
First PC	8859	199184	7819	37638	0.8207	0.1631	0.3722
Second PC	3519	190445	16558	42978	0.7651	0.0558	-0.0088
Third PC	3928	197725	9278	42569	0.7955	0.0704	0.1203
Fourth PC	2409	197068	9935	44088	0.7869	0.0427	0.0123
KTT-brightness	9750	201605	5398	36747	0.8337	0.1879	0.4895
KTT-greenness	8122	198718	8285	38375	0.8159	0.1483	0.3332
KTT-yellowness	8022	201363	5640	38475	0.8260	0.1539	0.4268
RVI	9409	199450	7553	37088	0.8239	0.1741	0.3979
NDVI	9387	199579	7424	37110	0.8243	0.1741	0.4016
TVI	9830	199268	7735	36667	0.8248	0.1813	0.4042
SAVI	8646	199113	7890	37851	0.8196	0.1590	0.3631
TDVI1	4173	205078	1925	42324	0.8254	0.0862	0.5132
TDVI2	6093	205327	1676	40404	0.8340	0.1265	0.6198
TDVI3	4173	205078	1925	42324	0.8254	0.0862	0.5132
TDVI4	6093	205327	1676	40404	0.8340	0.1265	0.6198
$c_2$	12282	2440180	31007	642317	0.7846	0.0179	0.0753

### 6.2.3. Texture Based Change Detection Methods

We provide the texture based change detection results in Table 6.6 using Otsu's method. For entropy method  $11 \times 11$  sliding window is used. For GLCM,  $13 \times 13$  sliding window used. As for GLCM features, Contrast, Energy and Inverse Difference Moment (IDM) are calculated. *PCC* ratings are lower than all pixel methods for all texture based methods. *Yule* ratings are also poor compared to pixel based methods. In terms of the *Jaccard* measure, IDM produced best result but with a very poor *PCC* and *Yule* rating.

Table 6.6. Performance measures for texture based methods using Otsu's method.

	TP	TN	FP	FN	PCC	Jaccard	Yule
Entropy	111029	2344253	126934	543570	0.7855	0.1421	0.2784
Contrast	179817	2190678	280509	474782	0.7584	0.1923	0.2125
Energy	244870	1978734	492453	409729	0.7114	0.2135	0.1606
IDM	276922	1933660	537527	377677	0.7072	0.2323	0.1766

### 6.2.4. Comparison of Thresholding Algorithms

Up to this point, we only provided change detection results for Otsu's method. Now, we evaluate the effect of Kapur's algorithm and percentile thresholding in change detection performance. Kapur's algorithm determines the threshold value automatically. However, a percent value should be determined for percentile thresholding. Therefore, we tested change detection methods starting from 99.0% to 75%. Decreasing the percentile value added more TP to the results. This led to an increase in the *Jaccard* measure. Rate of increase was high for the higher percentile values, and it gets lower when we drop the percentile to 75%. We demonstrate this behavior for image differencing method in Table 6.7. On this table, we focused on the *Jaccard* measure. As a reasonable level, we pick the percentile to be 85 % and applied it to all our methods.

We compared percentile thresholding, Otsu's method and Kapur's method using results from pixel based methods in Figure 6.4. We compared the *PCC*, *Jaccard* and *Yule* ratings produced by thresholding algorithms. If a thresholding algorithm produced better



Table 6.7. Effect of the percentile value on the performance measures.

Percentile (%)	TP	TN	FP	FN	PCC	Jaccard	Yule
99.0	17358	2465098	6089	637241	0.7942	0.0263	0.5349
97.5	43815	2451981	19206	610784	0.7985	0.0650	0.4958
95.0	86351	2425828	45359	568248	0.8037	0.1234	0.4658
90.0	164595	2360764	110423	490004	0.8079	0.2152	0.4266
85.0	231113	2279769	191418	423486	0.8033	0.2732	0.3903
80.0	286262	2184580	286607	368337	0.7905	0.3041	0.3554
75.0	365591	1954999	473481	283115	0.7541	0.3258	0.3092

*PCC*, *Jaccard* and *Yule* rating compared to another algorithm, then we say that algorithm performed better for the specific change detection method. Otsu's method performed better than Kapur's method when used for image differencing, image regression and CVA.

Thresholding algorithms are compared in terms of PCA and KTT methods in Figure 6.5. Otsu's method performed better than Kapur's method when used for KTT brightness and greenness bands.

We compared thresholding algorithms using the results from vegetation indices, TDVIs and color invariant in Figure 6.6. For RVI and NDVI, Kapur's method performed better than other algorithms. Otsu's method performed better than Kapur's algorithm when used with TVI and SAVI. Percentile thresholding performed better than Kapur's method for TDVI and color invariant.

We finally compared thresholding algorithms in terms of texture analysis based methods in Figure 6.7. Otsu's method performed better than Kapur's method when used with entropy based change detection. Percentile thresholding performed better than Otsu's method when used with contrast feature of the GLCM.

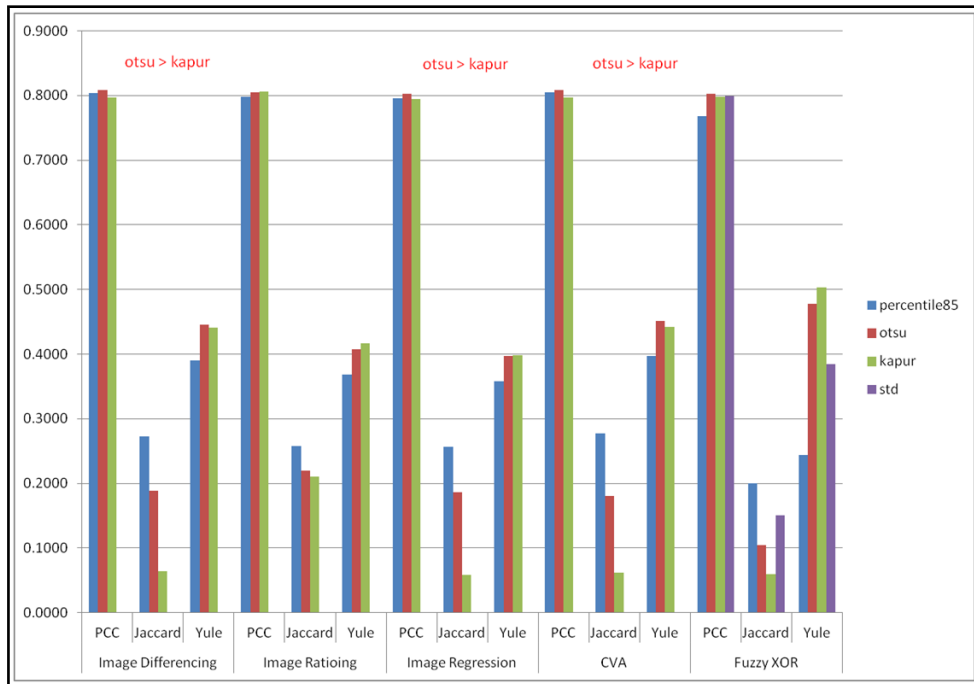


Figure 6.4. Thresholding algorithms are compared in terms of pixel based methods

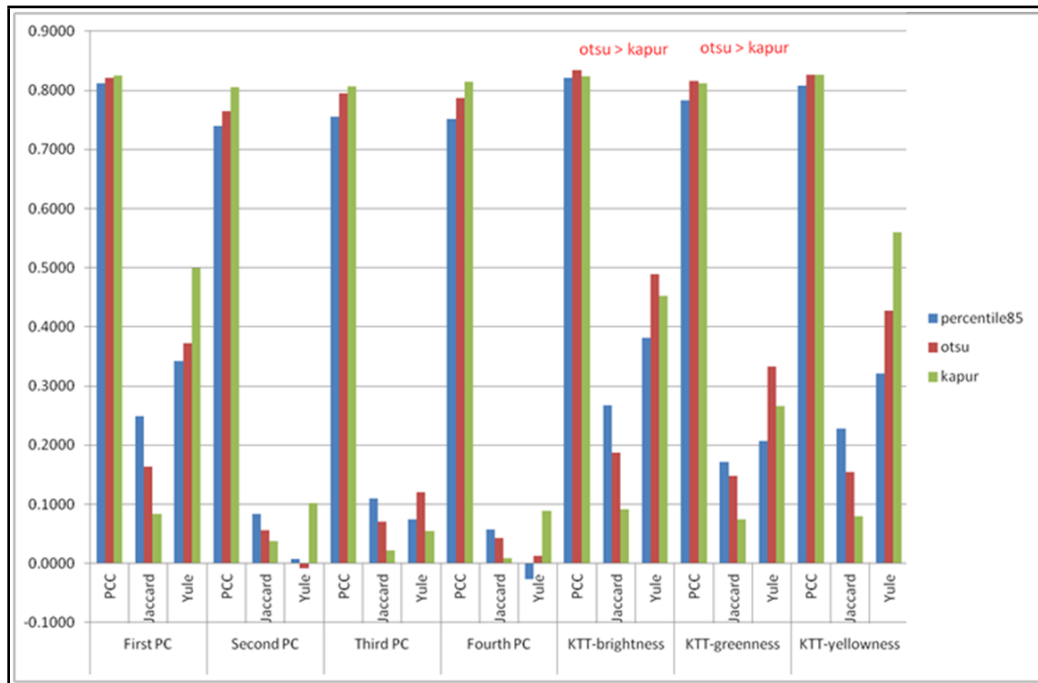


Figure 6.5. Thresholding algorithms are compared in terms of PCA and KTT

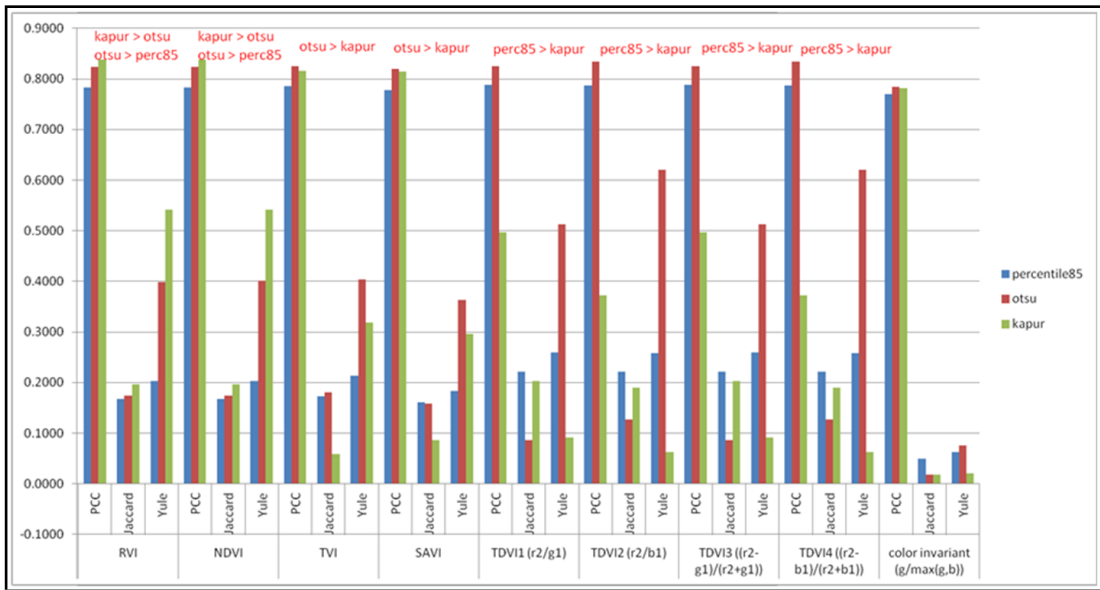


Figure 6.6. Thresholding algorithms are compared in terms of vegetation indices, TDVIs and color invariant

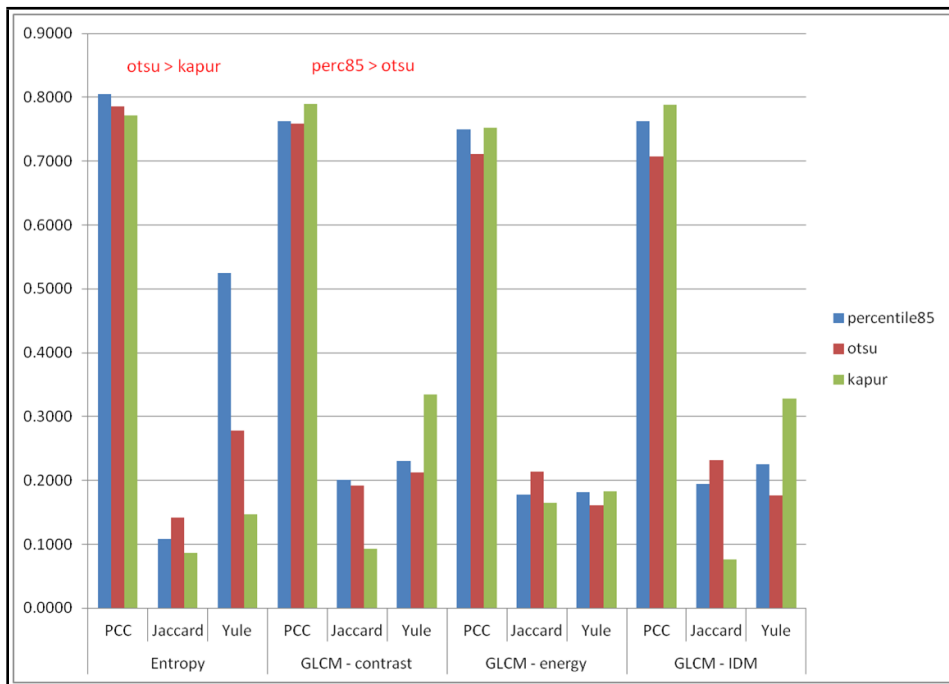


Figure 6.7. Thresholding algorithms are compared in terms of texture based change detection methods

### 6.2.5. Structure Based Change Detection Methods

So far we provided results based on the comparison of the methods to a ground truth in pixel basis. However, this can not be the case for structural methods, since every structural method generates different type of output. For example, edge based comparison produces thin lines representing object edges; graph matching based method produces only points; segment based method measures the change in segment basis. Thus, in structural method tests, we assume that smallest element a method's output can contain is an object. Objects are connected components which are set of spatially adjacent pixels. In order to handle object based comparison, we represented our ground truth database in a way that the ground truth is represented by unconnected objects. We removed some parts such as roads and areas between structures which connect several objects. Based on this definition, the number of objects in images  $AdanaP_1$  to  $AdanaP_8$  and  $AnkaraP_1$  to  $AnkaraP_{10}$  are [18, 10, 5, 14, 6, 9, 15, 14, 33, 75, 47, 46, 215, 113, 29, 21, 19, 28].

As we did for the pixel based methods, we need to find some comparison metrics between the ground truth set and the result set. We provide following statistics before we define our performance criteria for object based comparison. 'TP' is the number of truly found objects in the ground truth image. Objects are assumed to be truly found if any object in the result image overlaps a ground truth object. We also refer to these objects as matching objects. 'reTrue' is the number of objects in the result image which matches an object in the ground truth image. 'FN' is the number of objects in the ground truth image which are labeled as negative by the method. 'reFalse' is the number of objects in the result image which do not overlap any object in the ground truth image. Based on these measures, we benefit from two performance criteria named as Detection Percentage (DP) and Branching Factor (BF) used in [40]. These measures are defined as

$$DP = \frac{TP}{TP + FN} \quad (6.8)$$

$$BF = \frac{reTrue}{reTrue + reFalse} \quad (6.9)$$

Detection percentage is the comparison of the number of truly found objects to the number of ground truth objects. Branching factor decreases in proportion to the increase in the amount of result image objects which do not match any ground truth objects. Branching factor is an indication of increasing noise objects in the result image. A result image can obtain maximum detection percentage and branching factor by rendering whole image as true. In this case all ground truth objects are matched and since there is only one object in the result image, 'reTrue' is equal to 1 and 'reFalse' is equal to zero. As a result  $DP$  and  $BF$  are equal to their maximum value 1. In order to prevent this case we additionally, define the Pixel Correspondence (PCorr) measure

$$PCorr = \frac{rePxlOverlap}{rePxlTrue} \quad (6.10)$$

where 'rePxlOverlap' is the number of true pixels in the result image which overlap with an object in the ground truth image. 'rePxlTrue' is the number of true pixels in the result image. This measure gives an insight about the correspondence of the result image and the truth image. If pixel correspondence is very low, other criteria are assumed to be untrustful. In Table 6.8 we provide performance test results for the structure based change detection methods.

Table 6.8. Performance measures for the structure based methods.

Structure Type	TP	FN	reTrue	reFalse	DP	BF	Pcorr
Edge	314	403	419	643	0.4379	0.3945	0.2592
Matched Filtering	475	242	1028	2387	0.6625	0.3010	0.3854
GMSR	326	391	615	1291	0.4547	0.3227	0.3674
Segment	183	534	213	382	0.2552	0.3580	0.3257
Local Features	358	359	163	94	0.4993	0.6342	0.2980
Graph Matching	405	312	2099	2882	0.5649	0.4214	0.4713
Shadow	521	196	765	1956	0.7266	0.2811	0.1717

Shadow based comparison is best explained in terms of detection percentage. Pixel correspondence can not be used for interpreting shadow comparison results, because this method produces regions in the vicinity of the changed areas (not on them). When we

discard  $PCorr$  measure for shadow comparison, it has still problem. Low  $BF$  rating indicates high noise object production. Therefore, we can conclude that high  $DP$  rating is result of that. Matched filtering based comparison is produced second best  $DP$  rating. After that, comes graph matching based comparison. When we compare these two methods, even though matched filtering based method found more objects than graph based approach, it produced much noise and not as much associated as the graph matching based method to the ground truth. Graph matching based method has second high  $BF$  rating after local feature based comparison. Low correspondence of the local feature based approach depends on the large areas produced by the sum of the Gaussians as we discussed in Section 5.5. This method produces few large components which are low in association with the ground truth. Segment based approach produced poor detection results. Other methods produced moderate results. Eventually, graph matching based method produced high detection rate, has low noise and has good association to the ground truth in pixel basis.

## 7. CONCLUSIONS

In this thesis, we focused on change detection methods for satellite images. We grouped change detection methods under four categories based on the way they process data. We investigated several well-known methods which have already been applied and tested many times in the change detection domain. In addition to these methods, we provided novel structure based change analysis. We also measured the impact of three thresholding algorithms on the performance of change detection methods. We proposed a novel pixel based change detection method which accepts the gray scale image as a fuzzy set. Fuzzy XOR method produced results close to other pixel based methods. We also proposed two structure based change detection methods: local feature and graph matching based methods. Proposed structure based methods provide better result compared to other structure based methods. We also showed that, change detection is possible by comparison of shadow information for bitemporal images. Our performance evaluation was based on the comparison of the results from each method to a ground truth database composed of 35 image pairs. The ground truth database is created by visually comparing our data set. We provided comparison results in terms of three coefficients for both pixel based and object based comparison.

## REFERENCES

1. Singh, A., “Review Article Digital Change Detection Techniques Using Remotely-Sensed Data”, *International Journal of Remote Sensing*, Vol. 10, No. 6, pp. 989–1003, 1989.
2. Mas, J. F., “Monitoring Land-Cover Changes: A Comparison of Change Detection Techniques”, *International Journal of Remote Sensing*, Vol. 20, No. 1, pp. 139–152, 1999.
3. Lu, D., P. Mausel, E. Brondizio and E. Moran, “Change Detection Techniques”, *International Journal of Remote Sensing*, Vol. 25, No. 12, pp. 2365–2401, 2004.
4. Rosin, P. L. and E. Ioannidis, “Evaluation of Global Image Thresholding for Change Detection”, *Pattern Recognition Letters*, Vol. 24, No. 14, pp. 2345–2356, 2003.
5. Otsu, N., “A Threshold Selection Method from Gray-Level Histograms”, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 9, No. 1, pp. 62–66, 1979.
6. Kapur, J. N., P. K. Sahoo and A. K. C. Wong, “A New Method for Gray-Level Picture Thresholding Using the Entropy of the Histogram”, *Computer Vision, Graphics, and Image Processing*, Vol. 29, No. 3, pp. 273–285, 1985.
7. Griffiths, G. H., “Monitoring Urban Change from Landsat Tm and Spot Satellite Imagery by Image Differencing”, *Proceedings of the 1988 International Geoscience and Remote Sensing Symposium*, Vol. 1, 1988.
8. Saksa, T., J. Uuttera, T. Kolstrom, M. Lehtikoinen, A. Pekkarinen and V. Sarvi, “Clear-Cut Detection in Boreal Forest Aided by Remote Sensing”, *Scandinavian Journal of Forest Research*, Vol. 18, No. 6, pp. 537–546, 2003.
9. Lu, D., P. Mausel, M. Batistella and E. Moran, “Land-Cover Binary Change Detection



- Methods for Use in The Moist Tropical Region of The Amazon: A Comparative Study”, *International Journal of Remote Sensing*, Vol. 26, No. 1, pp. 101–114, 2005.
10. Malila, W. A., “Change Vector Analysis: An Approach for Detecting Forest Changes with Landsat”, *LARS Symposia*, p. 385, 1980.
  11. Johnson, R. D. and E. S. Kasischke, “Change Vector Analysis: A Technique for The Multispectral Monitoring of Land Cover and Condition”, *International Journal of Remote Sensing*, Vol. 19, No. 3, pp. 411–426, 1998.
  12. Klir, G. J. and B. Yuan, *Fuzzy Sets and Fuzzy Logic Theory and Applications*, Prentice Hall, 1995.
  13. Fung, T. and E. LeDrew, “Application of Principal Components Analysis to Change Detection”, *Photogrammetric Engineering and Remote Sensing*, Vol. 53, No. 12, pp. 1649–1658, 1987.
  14. Kauth, R. J. and G. S. Thomas, “The Tasseled Cap—A Graphic Description of The Spectral-Temporal Development of Agricultural Crops As Seen by Landsat”, *LARS Symposia*, p. 159, 1976.
  15. Seto, K. C., C. E. Woodcock, C. Song, X. Huang, J. Lu and R. K. Kaufmann, “Monitoring Land-Use Change in The Pearl River Delta Using Landsat Tm”, *International Journal of Remote Sensing*, Vol. 23, No. 10, pp. 1985–2004, 2002.
  16. Strang, G., *Linear Algebra and Its Applications*, Academic Press, 1976.
  17. Jordan, C. F., “Derivation of Leaf-Area Index From Quality of Light on the Forest Floor”, *Ecology*, pp. 663–666, 1969.
  18. Rouse, J. W., R. H. Haas and J. A. Schell, *Monitoring the Vernal Advancement and Retrogradation (Greenwave Effect) of Natural Vegetation*, Texas A and M University, 1974.

19. Jackson, R. D. and A. R. Huete, "Interpreting Vegetation Indices", *Preventive Veterinary Medicine*, Vol. 11, No. 3-4, pp. 185–200, 1991.
20. Perry Jr, C. R. and L. F. Lautenschlager, "Functional Equivalence of Spectral Vegetation Indices", *Remote Sensing of Environment*, Vol. 14, No. 1-3, pp. 169–182, 1984.
21. Huete, A. R., "A Soil-Adjusted Vegetation Index (Savi)", *Remote Sensing of Environment*, Vol. 25, No. 3, pp. 295–309, 1988.
22. Qi, J., A. Chehbouni, A. R. Huete, Y. H. Kerr and S. Sorooshian, "A Modified Soil Adjusted Vegetation Index", *Remote Sensing of Environment*, Vol. 48, No. 2, pp. 119–126, 1994.
23. Lunetta, R. S., J. F. Knight, J. Ediriwickrema, J. G. Lyon and L. D. Worthy, "Land-Cover Change Detection Using Multi-Temporal Modis Ndvi Data", *Remote Sensing of Environment*, Vol. 105, No. 2, pp. 142–154, 2006.
24. Guerra, F., H. Puig and R. Chaume, "The Forest-Savanna Dynamics from Multi-Date Landsat-Tm Data in Sierra Parima, Venezuela", *International Journal of Remote Sensing*, Vol. 19, No. 11, pp. 2061–2075, 1998.
25. Ünsalan, C., "Detecting Changes in Multispectral Satellite Images Using Time Dependent Angle Vegetation Indices", *3rd International Conference on Recent Advances in Space Technologies*, pp. 345 –348, June 2007.
26. Ünsalan, C. and K. L. Boyer, "Linearized Vegetation Indices Based on a Formal Statistical Framework", *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 42, No. 7, pp. 1575–1585, 2004.
27. Shafer, S. A., "Using Color to Separate Reflection Components", *Color Research and Application*, Vol. 10, No. 4, pp. 210–218, 1985.
28. Gevers, T. and A. Smeulders, "Color Based Object Recognition", *Image Analysis and*

*Processing*, pp. 319–326, 1997.

29. Haralick, R. M., K. Shanmugam and I. Dinstein, “Textural Features for Image Classification”, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 3, No. 6, pp. 610–621, 1973.
30. Tomowski, D., S. Klonus, M. Ehlers, U. Michel and P. Reinartz, “Change Visualization Through a Texture-Based Analysis Approach for Disaster Applications”, *Proceedings of ISPRS Technical Commission VII Symposium-100 Years ISPRS Advancing Remote Sensing Science Sensing*, Vol. 38, pp. 263–268, 2010.
31. Gonzalez, R. C., R. E. Woods and S. L. Eddins, *Digital Image Processing Using MATLAB*, Pearson Education, 2004.
32. Canny, J., “A Computational Approach to Edge Detection”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8, No. 6, pp. 679–698, 1986.
33. Sonka, M., V. Hlavac and R. Boyle, *Image Processing, Analysis, and Machine Vision*, Vol. 2, PWS Publishing, 1999.
34. Ünsalan, C., “Gradient-Magnitude-Based Support Regions in Structural Land Use Classification”, *IEEE Geoscience and Remote Sensing Letters*, Vol. 3, No. 4, pp. 546–550, 2006.
35. Stark, H. and J. W. Woods, *Probability and Random Processes with Applications to Signal Processing*, Vol. 3, Prentice Hall, 2002.
36. Comaniciu, D. and P. Meer, “Mean Shift: A Robust Approach Toward Feature Space Analysis”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 5, pp. 603–619, 2002.
37. Lowe, D. G., “Object Recognition from Local Scale-Invariant Features”, *The Seventh IEEE International Conference on Computer Vision*, Vol. 2, pp. 1150–1157, 1999.

38. Rosten, E. and T. Drummond, “Machine Learning for High-Speed Corner Detection”, *ECCV 2006*, pp. 430–443, 2006.
39. Romesburg, C., *Cluster Analysis for Researchers*, Lulu. com, 2004.
40. Lin, C. and R. Nevatia, “Building Detection and Description from a Single Intensity Image”, *Computer Vision and Image Understanding*, Vol. 72, No. 2, pp. 101–121, 1998.