Learning Texture Classifier for Flooded Region Detection in SAR Images

Shiqing Zhang, Hanqing Lu
National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
Beijing, China
email: {sqzhang,luhq}@nlpr.ia.ac.cn

Abstract

In this paper a new texture-based change detection approach is proposed to identify the flooded regions in SAR images. The main novelty of our approach is that the most distinctive texture information is automatically learned from the training set. Forty texture features, which are extracted from a pair of bi-temporal SAR images, are used to construct the weak classifier pool. After AdaBoost training, a strong classifier is optimally combined by a small subset of the candidate weak classifiers. The experimental results demonstrate the effectiveness of the proposed approach.

1. Introduction

The greenhouse effect, because of the continuous increase of the earth temperature, directly results in floods and other natural disasters. To monitor these disasters, image processing techniques could be used. But there usually exist heavy clouds on the flooded region, which may affect the image data acquisition of ordinary sensors. Therefore Synthetic Aperture Radar (SAR), which can capture qualified image data under different air and illumination conditions, is playing an important role in flood monitoring.

Usually the monitor of the flood is based on the detection of the changed water areas. There are several kinds of methods proposed for this kind of change detection. In several classical techniques, much attention has been paid on the difference image generated by calculating the absolute values of the difference between the corresponding pixels in two images. Large values in the difference image show the dissimilarity of the two areas and indicate changes, so thresholding the difference image at a specific value becomes a straightforward idea. Each pixel in the difference image, which has an intensity value that is larger than a threshold, is labeled as changed, while the rest are considered to be unchanged. Although several methods [1] have been proposed to choose an appropriate threshold, thresholding is still sensitive to the noise and illumination changes and as a result not suitable for applications in complex environments, for example detecting changes of waters in SAR images.

In recent years, more and more efforts have been made to exploit the texture information contained in SAR images. Rather than considering information restricted within a pixel, texture features are extracted from pixels in a region, which are less sensitive to the noise that is very common in remote sensing images. Many change detection methods [2, 3, 4] presented recently are based on the texture differences between two images. Although these methods vary from one to another, they all tend to use more texture features than before to obtain higher accuracy. In fact there are some drawbacks to use too many texture features. Usually more texture features mean more information, but too many texture features will not only increase the complexity of computation but also place a heavy burden on selecting the appropriate ones used in classification. How to select a minimum subset of the features, which is suitable for change detection applications, becomes an open problem. Traditional feature selection methods such as PCA, ICA cannot maximize interclass separability in the lower dimensional subspace [5]. In our work the AdaBoost algorithm, which has been proved to be very successful in selecting features in face detection and recognition problems, is employed to choose the texture features with the best discriminability from all the forty ones. Based on these selected textures we build up a strong classifier to detect the flooded areas in the bi-temporal SAR images.

This paper is organized as follows: In section 2 we will give an overlook of the whole work. Section 3 will give a detailed description of the texture features used in our work. We will present the AdaBoost algorithm and the strong classifier used in change detection in section 4. Experimental results and comparison are given in section 5. Finally, this paper is concluded in section 6 with discussions.

2. Overlook of the work

To detect the flooded areas in two SAR images, our flooded areas detection algorithm(see Figure 1) is made up of five steps: 1)preprocessing of the SAR images, 2)texture feature extraction, 3)weak classifier design, 4)construction of the strong classifier using the AdaBoost algorithm, 5)change detection and identification of flooded regions.

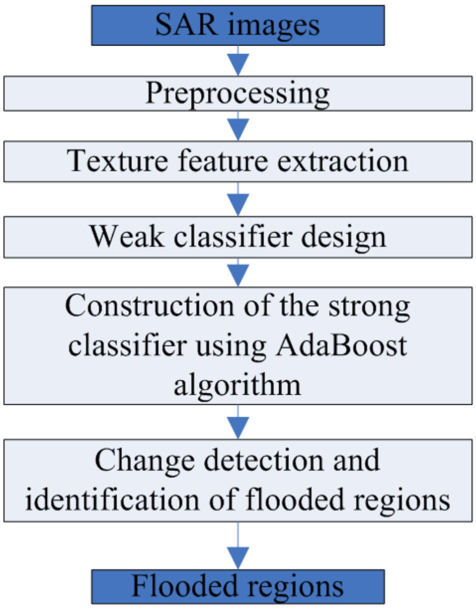

SAR images
Preprocessing
Texture feature extraction
Weak classifier design
Construction of the strong classifier using AdaBoost algorithm
Change detection and identification of flooded regions
Flooded regions

Figure 1. Flooded region detection algorithm.

- **Preprocessing of the SAR images:** Because registration and radiometric normalization have been done in previous work, we only need to concentrate on the speckle noise in the SAR images. In our work Most Homogeneous Region Filter [7] is adopted to reduce the speckle noise.
- **Texture feature extraction:** In this step we define four kind texture features to detect changes in the bi-temporal SAR images. They are the Kullback-Leibler distance between the local neighborhood pixel intensity distributions, the mean, variance, and median differences of the local neighborhood pixel intensity. Because we do not know the best neighborhood window size for the change detection application, each kind of texture feature is extracted with ten different window sizes, ranging from 3×3, 5×5,$\cdots$, to 21×21. Therefore forty texture features are extracted from the bi-temporal SAR images.
- **Weak classifier design:** For each texture feature we get from two SAR images, a weak classifier is designed. We call the classifier a ”weak” one because we do not expect high accuracy. Thresholding is adopted to make a rough separation of the changed and unchanged pixels.
- **Construction of the strong classifier using the AdaBoost algorithm:** The basic idea of boosting is to combine several weak classifiers to a strong one. In this step, the AdaBoost algorithm is employed to choose several suitable classifiers from all the forty weak ones and the final strong classifier is made up of a linear combination of them.
- **Change detection and identification of flooded regions:** Using the strong classifier learned from the AdaBoost algorithm, the change detection result is satisfying. But not all of the changes are made by the flood, some of them can be due to other factors. In order to gain a desired result, postprocessing should be made to validate the real inundated regions. In this step, two prior-knowledge based strategies are adopted to identify the final result.

3 Texture feature description

In SAR images water areas and land areas are usually quite different. There are two major differences between water areas and land areas, brightness and roughness. Water areas are much darker and smoother than land areas. So in changed areas both the brightness and the roughness of the areas vary a lot, while in unchanged areas there are not very significant differences between two images. We found that the mean, variance, and median of the pixel intensity in the spatial neighborhood in the bi-temporal SAR images can express the differences of brightness and roughness.
For a given point $p = (u, v)$ in the image, let ω_N denotes the $N \times N$ spatial neighborhood. The N^2 points within the neighborhood of the ith image are labeled as $\{x_1^{(i)}, x_2^{(i)}, \cdots, x_{N^2-1}^{(i)}, x_{N^2}^{(i)}\}$, where $i \in \{1, 2\}$. The mean, variance and median of the pixel intensity can be written as

$$\mu_i = \frac{1}{N^2}\sum_{j=1}^{N^2} x_j^{(i)}, \tag{1}$$

$$\sigma_i^2 = \frac{1}{N^2}\sum_{j=1}^{N^2}(x_j^{(i)} - \mu_i)^2, \tag{2}$$

$$med_i = median_{j\in(1,2,\cdots,N^2)}\{x_j^{(i)}\}. \tag{3}$$

To describe the texture difference, a measurement of the difference is defined as

$$D(f_1, f_2) = 1 - \frac{2{\cdot}f_1{\cdot}f_2}{f_1^2 + f_2^2}, \tag{4}$$

where$f_i \in \{\mu_i, \sigma_i^2, med_i\}, i \in \{1, 2\}$. It is obvious that $D(f_1, f_2)$ equate to 0 when $f_1 = f_2$, which shows the similarity between f_1 and f_2, while $D(f_1, f_2)$ reaches a much larger value when $f_1 \gg f_2$ or $f_2 \gg f_1$, which indicates a difference.

In order to make full use of the information in each pixel's local neighborhood, we take into consideration the statistical distribution of the intensity histogram in the neighborhood and choose a measurement called Kullback-Leibler distance [12] to describe the difference.

Let H_1 and H_2 be the intensity histogram of the $N \times N$ spatial neighborhood in two SAR images. Let P_1 and P_2 be the discrete probability laws of H_1 and H_2 . The Kullback-Leibler divergence is defined as

$$K(P_1, P_2) = \sum_i log \frac{P_2(\{x_i\})}{P_1(\{x_i\})} P_2(\{x_i\}). \tag{5}$$

One may notice that in most of the time $K(P_1, P_2) \neq K(P_2, P_1)$, in other words, the Kullback-Leibler divergence is not symmetric. A symmetric distance can be achieved by defining the Kullback-Leibler distance as

$$D(P_1, P_2) = K(P_1, P_2) + K(P_2, P_1). \tag{6}$$

Therefore four kind texture features are extracted, $D(\mu_1, \mu_2)$, $D(\sigma_1, \sigma_2)$, $D(med_1, med_2)$, $D(P_1, P_2)$. Because for each kind of texture feature we do not know the most suitable neighborhood window size that can lead to the best discriminability, four texture features are extracted with different neighborhood window sizes. In our work ten kinds of window sizes are adopted, ranging from 3×3, $5 \times 5, \cdots$, to 21×21. Finally forty texture features are extracted from two SAR images.

4 Methodology

4.1 The AdaBoost algorithm

The intrinsic purpose of boosting is to combine several weak classifies to a strong one that can get better accuracy than any weak one. The AdaBoost algorithm [9], a sequential forward search procedure using the greedy selection strategy, has overcome some practical difficulties that earlier ones hold [13].

```
1. Input:
   $S = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$
   where $y_i \in \{+1, -1\}$
   number of the iteration $T$

2. Initialization:
   $d_i^{(1)} = \frac{1}{N}$      for all $i = 1, 2, \cdots, N$

3. Iteration:
   For $t = 1, \cdots, T$
     • Obtain weak classifier $h_t : x \to \{+1, -1\}$
     • Calculate the weight training errors $\epsilon_t$ of $h_t$:
       $\epsilon_t = \sum_{i=1}^{N} d_i^{(t)} I(y_i \neq h_t(x_i))$
     • Update weights:
       $\alpha_t' = \frac{1}{2} log \frac{1-\epsilon_t}{\epsilon_t}$
       $d_i^{(t+1)\prime} = d_i^{(t)} exp\{-\alpha_t' y_i h_t(x_i)\}$
       $d_i^{(t+1)} = \frac{d_i^{(t+1)\prime}}{\sum_{i=1}^{N} d_i^{(t+1)\prime}}$
     • Break when $\epsilon_t = 0$ or $\epsilon_t \geq \frac{1}{2}$, set $T = t - 1$,
       $\alpha_t = \frac{\alpha_t'}{\sum_{t=1}^{N} \alpha_t'}$

4. Output:
   $H(x) = sign[\sum_{t=1}^{T} \alpha_t h_t(x)]$
```

Figure 2. The AdaBoost algorithm [8].

For a two class classification problem, initially we have a training set S of N examples $(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)$, where $y_i \in \{+1, -1\}$ is the class label of the corresponding examples . In our approach "1" stands for change and "-1" denotes no change. In AdaBoost algorithm (see Figure 2) each example (x_i, y_i) is associated with a weight defined as $d_i^{(t)}$ at the *t*th iteration. At each iteration step *t*, examples that are misclassified get higher weights in the next iteration according to the error incurred by the weak learner h_t, in other words more attention is paid on the examples that are hard to classify. The weak classifier is required to produce a small weighted empirical error defined by

$$\epsilon_t(h_t, d^{(t)}) = \sum_{i=1}^{N} d_i^{(t)} I(y_i \neq h_t(x_i)). \tag{7}$$

After selecting the weak classifier h_t , we have to update its weight α_t to minimize the loss function defined as

$$G(\alpha_t) = \sum_{i=1}^{N} exp\{-y_i(\sum_{j=1}^{t} \alpha_j h_j(x_j))\}. \tag{8}$$

A strong classifier that is made up of a linear combination of several weak classifiers can be defined as

$$H_T(x) = \sum_{t=1}^{T} \alpha_t h_t(x). \tag{9}$$

The final classifier is presented as

$$H(x) = sign[H_T(x)]. \tag{10}$$

4.2 Weak classifier design

In our work forty texture features are extracted from two SAR images, consequently we have to design a classifier for each of them. It is found that both the Kullback-Leibler distance and the difference of the mean, variance and median get a low value when there is no change while a high one when there is a change. Because very high accuracy is not necessary in weak classifier design, thresholding, a simple approach with relatively low computation complexity, is adopted in weak classifier design. For the *i*th texture feature a threshold thr_i is obtained by minimize the error rate in the training examples. The corresponding weak classifier $h_i(x)$ is defined as:

$$h_i(x) = \begin{cases} +1 & \text{if } x > thr_i \\ -1 & \text{if } x \leq thr_i \end{cases} . \qquad (11)$$

4.3 Change detection implementation

The training set of the AdaBoost algorithm is two SAR images with a size of 512×512 and a corresponding ground-true image of the changed areas used as a reference map. For each pair of points having the same coordinate in two images, forty texture features are extracted from the bi-temporal SAR images. Each weak classifier is constructed based on a single texture feature to obtain a minimum error rate in training set. After that we employ the AdaBoost algorithm to combine weak classifiers into a strong one, which will be used in change detection. For the test set we also need to do the texture extraction work and use the strong classifier we got to label each pixel as either changed or unchanged.

4.4 Identification of flooded regions

In last step, a strong classifier has been implemented by using the AdaBoost algorithm and the change detection result (In our work it is presented as a change image.) is achieved with the help of that classifier. But it is obvious that not all the changes in the image are made by the flood. Some of them are made by other reasons, which we are not interested in. Several strategies have been proposed to identify the real inundated regions. In our work, two prior-knowledge based approaches [6] have been adopted.

First, we are interested in changed areas with relative large size, so too small regions are eliminated from the change image. That is because connected regions with very small amount of pixels and some single isolated pixels are more likely be done by the noise that is very prevalent in SAR images.

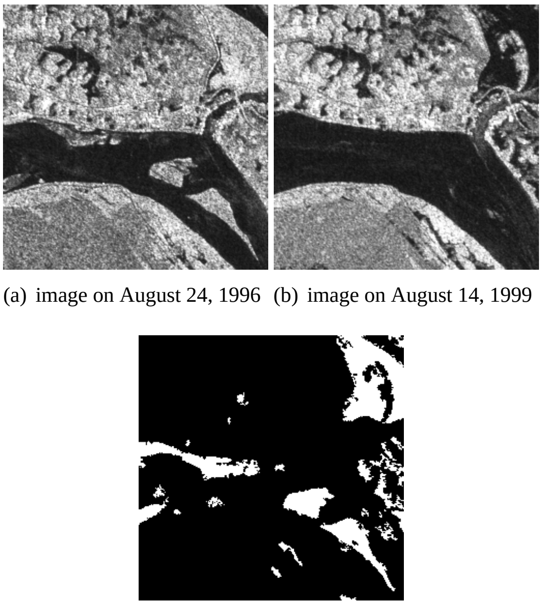
Second, it is common knowledge that flooded regions were land area before the flood and were water areas when the flood happened. As we know water areas are darker and smoother that land areas, both the mean and variance of the intensity values in the flooded regions should move from low to high. In our work regions that do not change from land to water are excluded from the final identification result of the flooded regions.

5 Experimental results

5.1 Data description

In order to evaluate the performance of the method proposed in this paper, we consider two pairs of bi-temporal SAR images of Red River, Vietnam on August 24, 1996 and August 14, 1999. One is used as training set while another is used as test set. Since registration and radiometric normalization have been done in previous work, we only need to focus on denoising. Most Homogeneous Region Filter [7] is adopted to reduce the speckle noise.

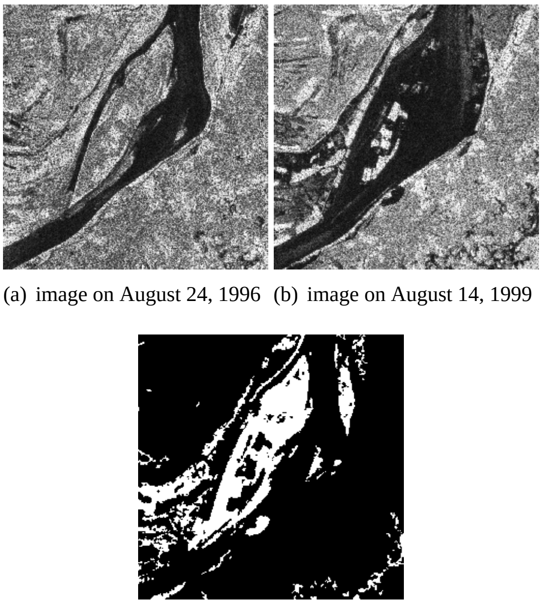
We use one pair of images, both of which have a size of 512×512, in our training step. The changed and unchanged areas are labeled by a manual analysis of the two SAR images to form a reference map. The two SAR images and the ground-truth map of the changed and unchanged areas used as the training set are shown in Figure 3. The test set consisting of another pair of SAR images, which have a size of 680×680, and the ground-truth map is shown in Figure 4.



(a) image on August 24, 1996 (b) image on August 14, 1999

(c) ground-truth map

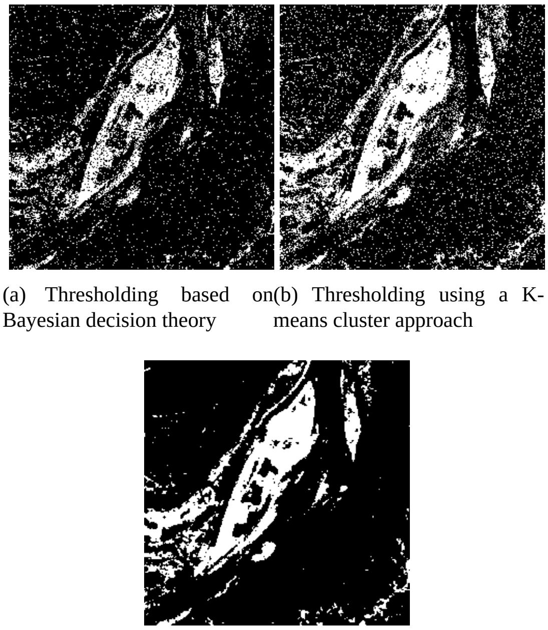
Figure 3. Images of Red River used as training set.



(a) image on August 24, 1996 (b) image on August 14, 1999

(c) ground-truth map

Figure 4. Images of Red River used as test set.



(a) Thresholding based on Bayesian decision theory (b) Thresholding using a K-means cluster approach

(c) Our approach

Figure 5. Change detection result comparison.

Table 1. The qualitative comparison of the proposed method with two thresholding methods.

Method	False alarms	Miss alarms	Overall errors	Overall errors(%)
Method*	18841	23053	41894	9.06
Method**	44137	7960	52097	11.27
Proposed	5356	751	6107	1.32

Method* : Thresholding based on Bayesian decision theory.
Method**: Thresholding using a K-means cluster approach.

5.2 Change detection result and performance comparison

In all the forty texture features, five of them are selected by using the AdaBoost algorithm and consequently the strong classifier is constructed by a linear combination of the five weak classifiers. In order to show that our method is more robust to the noise that is very prevalent in SAR images, change detection results of two methods based on thresholding as well as the result of our approach are shown Figure 5. One method proposed in [10] detects changes in the difference image by a global threshold in terms of the Bayesian decision theory. Another method [11] adopts a K-means cluster approach to choose the appropriate threshold. It is obvious that the proposed method is less sensitive to the noise and results in a more reliable outcome.

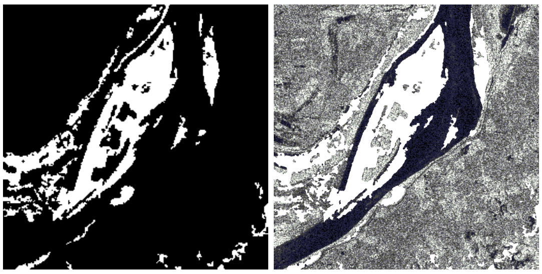
The qualitative analysis of the accuracy provided by proposed method is compared with that exhibited by the two thresholding approaches in Table 1. The table gives the numbers of the false alarms (i.e. unchanged pixels classified as changed ones) and the miss alarms (i.e. changed pixels classified as unchanged ones) brought by both our method and two thresholding approaches. Apparently our method achieves a significant improvement. The combination of the selected weak classifier provides an accurate classification of the changed areas and the result is less susceptive to the noise.

5.3 Flooded regions extraction result

For we are only interested in changed areas that are made by the flood, changed regions caused by other factors are removed by using the two strategies mentioned in section 4. The final identification result of the inundated regions is show in Figure 6.

6 Conclusions

This paper proposes a new change detection algorithm based on multi-texture information to identify the flooded



(a) flooded regions map

(b) flooded regions map to the original non-flooded image (white: flooded regions)

Figure 6. Flooded regions extraction result.

regions in SAR images. Unlike traditional change detection methods, which rely only on a limited amount of texture features because of the difficulty in selecting suitable ones with the best discriminability, forty texture features are extracted from every two SAR images and are used to construct the weak classifier pool. Instead of using PCA or ICA, which cannot lead to a maximum interclass separability in the lower dimensional subspace, we adopt the AdaBoost algorithm, which has been proved to be quite effective in selecting features in face detection and recognition problems and has hardly been used in remote sensing change detection, in our method. With the help of the AdaBoost algorithm we build up a strong classifier combined by a small subset of the candidate weak classifiers to detect the flooded areas in the bi-temporal SAR images.
Experimental results show that the classifier is quite efficacious in detecting changes in SAR images and confirm the validity of the proposed approach. Although the result is rather encouraging, some further work can still be done. For example, in our work, texture features are based on the basic statistics of the local pixel intensity. In fact more texture features, such as some wavelet-based texture features, can be added into our method and may bring an even better performance.

References

[1] Paul L. Rosin, Thresholding for change detection, *Proc. 6th International Conference on Computer Vision*, Bombay, India, 1998, 274-279.

[2] Liyuan Li and Maylor K.H. Leung, Integrating intensity and texture differences for robust change detection, *IEEE Transactions on Image Processing*, 11(2), 2002, 105-112.

[3] Liyuan Li and Maylor K.H. Leung, Robust change detection by fusing intensity and texture differences, *Proc. 4th Computer Vision and Pattern Recognition*, 2001, 777-784.

[4] Mohamed Borchani and Florence Cloppet and Volkan. Atalay and Georges. Stamon, Change detection in aerial images, *Proc. 1st Canadian Conference on Computer and Robot Vision*, Victoria, Canada, 2004, 354-360.

[5] Sergios Theodoridis and Konstantinos Koutroumbas, *Pattern Recognition Second Edition*, (Amsterdam: Academic Press, 2003).

[6] Wenhui Peng, Chunhong Pan and Veronique Prinet, Unsupervised change detection for flood analysis in SAR images, *Proc. 6th Asia Conference on Computer Vision*, Jeju Island, Korea, 2004, 652-657.

[7] Yifeng Wu and Henri Affiliation Maitre, Smoothing speckled Synthetic Aperture Radar images by using maximum homogeneous region filters, *Optical Engineering*, 31(8), 1992, 1785-1792.

[8] Yoav Freund and Robert E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, *Journal of Computer and System Sciences*, 55(1), 1997, 119-139.

[9] Ron Meir, Gunnar and Rätsch, An introduction to boosting and leveraging, Lecture Notes In *Artificial Intelligence Advanced lectures on machine learning*, 2003, 118-183.

[10] Lorenzo Bruzzone and Diego Fernàndez Prieto, Automatic analysis of the difference image for unsupervised change detection, *IEEE Transactions on Geoscience and Remote Sensing*, 38(3), 2000, 1171-1182.

[11] Fangju Wang, A knowledge-based vision system for detecting land changes at urban fringes, *IEEE Transactions on Geoscience and Remote Sensing*, 31(1), 1993, 136-145.

[12] Jordi Inglada, Change detection on SAR images by using a parametric estimation of the Kullback-Leibler divergence, *Proc. 29th Geoscience and Remote Sensing Symposium, IGARSS*, Toulouse, France, 2003, 4104-4106.

[13] Stan Z. Li, Long Zhu, ZhenQiu Zhang, Andrew Blake, HongJiang Zhang and Harry Shum, Statistical learning of multi-view face detection, *Proc. 7th European Conference on Computer Vision*, Copenhagen, Denmark, 2002, 67-81.