Fast Object-Level Change Detection for VHR Images

Chunlei Huo, Zhixin Zhou, Hanqing Lu, *Member, IEEE*, Chunhong Pan, and Keming Chen

***Abstract*—A novel approach is presented for change detection of very high resolution images, which is accomplished by fast object-level change feature extraction and progressive change feature classification. Object-level change feature is helpful for improving the discriminability between the changed class and the unchanged class. Progressive change feature classification helps improve the accuracy and the degree of automation, which is implemented by dynamically adjusting the training samples and gradually tuning the separating hyperplane. Experiments demonstrate the effectiveness of the proposed approach.**

***Index Terms*—Fast multitemporal segmentation, object-level change vector analysis, progressive classification.**

I. Introduction

COMPARED with low- to moderate-resolution remote sensing images, change detection of very high resolution (VHR) images is more important since it can detect changes at the more detailed spatial scale. However, change detection of VHR images is more difficult mainly due to the following factors. First, the discriminability between different land-cover classes is determined simultaneously by the spatial resolution and the spectral resolution. Although the spatial resolution of VHR sensors increases significantly, restricted by the technical constraints (there is a tradeoff between the spatial resolution and the spectral resolution [1]), the spectral resolution and the spectral capabilities are relatively poor. As a result, the overall separability between the changed class and the unchanged class is not improved simultaneously with the increased spatial resolution. Second, traditional pixel-based approaches are limited for VHR images [2], and object-level change detection approaches have aroused great attention recently. However, due to the complex statistical distributions of the changed class and the unchanged class, most existing approaches classify the object-based change features by setting the threshold empirically, which is time-consuming and lacks of generalizability. Third, most techniques ignore the dependence information between two images [3]. Due to the ill-defined nature of the change detection problem (for example, the changes of interest are mixed up with normal changes), the changed class is difficult to be separated from the unchanged class purely based on the "relative" change features.

The existing techniques are far from satisfied with the practical requirements from applications, and it is mandatory to invest further efforts to develop effective solutions. To overcome the aforementioned difficulties, in this letter, a novel change detection approach is presented for VHR images. Compared with related techniques, the proposed approach is promising in automation, robustness, effectiveness, and efficiency.

This letter is organized as follows. The proposed approach is elaborated in detail in Section II. The experiments are reported in Section III. Finally, the conclusions are given in Section IV.

II. Fast Object-Level Change Detection Based on Progressive Classification

The rationale of the proposed approach is to extract object-specific regions by fast multitemporal segmentation, to represent the object-specific difference by combining the relative change features and the original multitemporal spectral features, and to classify the object-specific change features automatically by progressive classification. Next, we will elaborate the proposed approach step by step.

A. Fast Multitemporal Segmentation

The minimized unit of object-level change detection approaches is the object, which is extracted by segmentation. Among the different segmentation approaches, the graph-based segmentation approach presented by Felzenszwalb and Huttenlocher [4] is chosen in this letter due to its promising performances [5]. For example, this algorithm is reliable and stable (segmentation is neither too coarse nor too fine [4]), automatic (three parameters need to be specified, and segmentation is less sensitive to these parameters [5]), and fast (for a 1024×1024 image, the CPU time is about 20 s on a personal computer). The reader is referred to [4] and [5] for the details of the algorithm.

To take advantage of the object-level approach, we first apply the aforementioned segmentation procedure on the two coregistered images separately. To avoid the shortcoming of the postclassification-based change detection approach, the multitemporal segmentation is then applied. For a region B in one image and the corresponding region A in the other image, the final segmentation is achieved according to the following rules.

1) If A has no subregions and $A = B$, then A keeps unchanged in the final result.
2) If $A = B$ and A is composed of subregions, i.e., $A = A_1 \cup \cdots \cup A_n$, then the final result in $A \cup B$ is split into n regions: $A_1, \ldots, A_n$.
3) If $A \neq B$ and $A \bigcap B \neq \Phi$, then the region $A \cup B$ in the final result is split into three regions: $A \bigcap B$, $A - (A \bigcap B)$, and $B - (A \bigcap B)$.

Manuscript received October 13, 2008; revised February 8, 2009 and June 9, 2009. First published September 18, 2009; current version published January 13, 2010. This work was supported by the Natural Science Foundation of China under Grant 60605004.

C. Huo, H. Lu, C. Pan, and K. Chen are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: clhuo@nlpr.ia.ac.cn; hqlu@nlpr.ia.ac.cn; chpan@nlpr.ia.ac.cn; kmchen@nlpr.ia.ac.cn).

Z. Zhou is with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with Beijing Institute of Remote Sensing, Beijing 100854, China (e-mail: zxzhou@nlpr.ia.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LGRS.2009.2028438

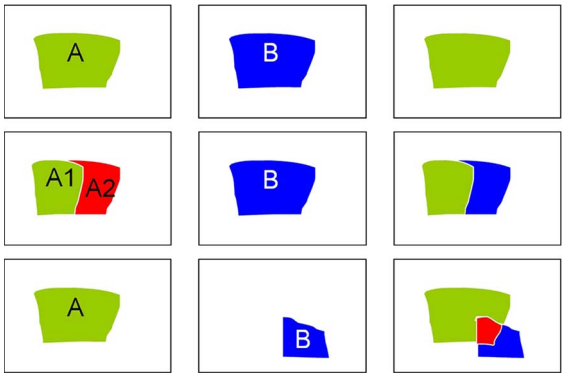


Fig. 1. Illustration of multitemporal segmentation. (First row) Case (1). (Second row) Case (2). (Third row) Case (3). (First column) Segmentation result from one image. (Second column) Segmentation result from the other image. (Third column) Final segmentation result.

The aforementioned procedure is illustrated in Fig. 1. Based on the aforementioned fast multitemporal segmentation, the homogeneous regions are extracted, and the interclass separability is also improved.

B. Object-Based Change Feature Extraction

Change vector analysis [6] is very effective in combining the change features of different kinds (or different spectral bands). However, traditional change vector analysis is implemented in a pixel-based fashion, which is susceptible to noise and time-consuming for the later classification. For this reason, we generalize the concept of change vector analysis and propose an object-based change vector analysis. For coregistered multitemporal images I_1 and I_2, the object-based change magnitude within region R_j can be represented by D_j

$$D_j = \frac{1}{\|R_j\| \times C} \sqrt{\sum_{i=1}^{C} \sum_{(x,y)\in R_j} \left(I_1^{(i)}(x,y) - I_2^{(i)}(x,y) \right)^2} \quad (1)$$

where C is the number of spectral bands, $I_1^{(i)}$ is the component of I_1 at the ith band, and $\|R_j\|$ is the number of pixels within region R_j. In general, the relative change feature D_j (1) is not discriminable enough to represent the complex changes contained in the VHR images. To reflect the dependence information between the images, the original multitemporal signatures are added to the change features, i.e.,

$$\mathbf{x}_i = \left[I_1^{(1)}(R_i), \ldots, I_1^{(C)}(R_i), I_2^{(1)}(R_i), \ldots, I_2^{(C)}(R_i), D_i \right]. \quad (2)$$

$I_k^{(j)}(R_i)$ is the mean value of I_k within R_i at band j.

C. Progressive Change Feature Classification

Considering the complex statistical distributions of the changed class and the unchanged class, a support vector machine (SVM), a distribution-free classifier, is used to classify the object-specific change features. The other reason for adding the original multitemporal signatures into the change feature vectors is the powerful ability of the SVM in processing high-dimensional data. Despite the advantages of the SVM in classification, it requires training samples labeled beforehand, which limits the application of the SVM for automatic change detection. In this letter, we address this problem by progressive classification, which is composed of two steps: initial classification and refined classification. In initial classification, some potential training samples are selected automatically, and the initial change map is determined by the initial separating hyperplane. In the refined procedure, the performance is improved gradually by adjusting the training samples dynamically.

1) Initial Classification: The change feature D_i (1) indicates the degree of changes within region R_i; the higher D_i is, the more probability of changes will be. Therefore, the sample set $S^c = \{(\mathbf{x}_i, y_i)\}$ ($y_i = 1$, changed class) with correspondence to the first $\alpha \times N$ (e.g., $0.2 \times N$) largest part of $D = \{D_i | i = 1, \ldots, N\}$ can be regarded as the positive training samples, and $S^u = \{(\mathbf{x}_i, y_i)\}$ ($y_i = -1$, unchanged class) with correspondence to the first $\alpha \times N$ smallest part of D can be regarded as the negative training samples. The initial classifier is obtained by the inductive SVM (*iSVM*) on the training set $S_{\text{train}} = S^c \bigcup S^u$, and the initial change map can be achieved by computing the decision function values on all the unlabeled examples.

Due to the uncertainty of the training samples and the inherent nature of the *iSVM*, the initial change map may not be accurate enough. The uncertainty of the training samples lies in the fact that the training samples are based only on the relative change cues, and they are limited to represent the whole change feature set. The inherent nature of the *iSVM* lies in the purpose of optimizing the classification performance over all the possible future test data. In fact, this is not necessary since we are only interested in the features extracted from the images being considered. Consequently, the classification accuracy is refined by progressive classification, which is implemented by the iterative transductive SVM (*tSVM*).

2) Refined Classification: Given a set of independent labeled examples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$, $y_i \in \{-1, +1\}$, and unlabeled examples $\mathbf{x}_1^*, \ldots, \mathbf{x}_k^*$ from the same distribution, the aim of the *tSVM* [7] is to minimize the following equation over $(y_1^{*\prime}, \ldots, y_k^*, w, b, \xi_1, \ldots, \xi_n, \xi_1^*, \ldots, \xi_k^*)$:

$$\min \left(\frac{1}{2}\|w\|^2 + C \sum_{i=1}^{n} \xi_i + C^* \sum_{j=1}^{k} \xi_j^* \right) \quad (3)$$

$$\text{s.t.} \quad \forall_{i=1}^{n} : \; y_i \left(w \cdot \phi(\mathbf{x}_i) + b \right) \le 1 - \xi_i, \quad \xi_i \ge 0 \quad (4)$$

$$\forall_{j=1}^{k} : \; y_j^* \left(w \cdot \phi\left(\mathbf{x}_j^*\right) + b \right) \le 1 - \xi_j^*, \quad \xi_j^* \ge 0 \quad (5)$$

where the regularization parameters C and C^* control the generalization capabilities, ξ_i and ξ_j^* are the positive slack variables enabling to deal with the permitted errors, and ϕ is the mapping function. For the change detection algorithm based on the iterative *tSVM*, at each iteration, we remove the nonrepresentative training examples from the training set and add the new representative examples from the unlabeled examples to the training set. Based on the hyperplane $f(\mathbf{x}_i) = \sum_j \alpha_j y_j K(\mathbf{x}_j, \mathbf{x}_i) + b = \sum_j \alpha_j y_j < \phi(\mathbf{x}_j), \phi(\mathbf{x}_i) > + b$ and Karush–Kuhn–Tucker condition, the training examples $\mathbf{x}_i$ can be partitioned into three different categories according to $g_i = y_i f(\mathbf{x}_i) - 1$ [8]: the set S of margin support vectors strictly on the margin ($g_i = 0$), the set E of error support vectors exceeding the margin ($g_i < 0$), and the remaining set R of

reserve vectors exceeding the margin ($g_i > 0$). At the next iteration, the set E should be deleted from the training set since its label is inconsistent with the current separating hyperplane, and it contradicts the assumption that the objective function has been minimized. For the unlabeled examples $\mathbf{x}_1^*, \ldots, \mathbf{x}_k^*$, only those lying within the margin band are important for the later classification since the adding of them to the training set may change the separating hyperplane. To keep the whole classification stable, at each iteration, we add the new representative training examples in a pairwise manner (one representative positive example and one representative negative example) to the training set whenever possible. In this context, the "representative" examples are the ones that lie within the margin band and are closest to the margin. However, if there is no such pair, only one representative positive or negative example is picked and added to the training set. The iteration continues unless all the unlabeled examples are outside the margin band. By this way, the accuracy is improved iteratively by dynamically selecting the "representative" training samples and gradually tuning the separating hyperplane.

For convenience, the representative positive and representative negative examples are described as $S_{\text{add}}^{+} = \{(\mathbf{x}_i^*, y_i^*) | i = \arg \max_{j:0<f(\mathbf{x}_j^*)<1} |f(\mathbf{x}_j^*)|\}$ and $S_{\text{add}}^{-} = \{(\mathbf{x}_i^*, y_i^*) | i = \arg \max_{j:-1<f(\mathbf{x}_j^*)<0} |f(\mathbf{x}_j^*)|\}$, respectively. The progressive change feature classification can then be described as follows:

1. Initial classification.

1.1 Sort D in ascending order; select the initial training set S_{train}.

1.2 Train the classifier by applying the *iSVM* on the training samples, and achieve the initial change map by classifying the whole feature set.

2. Refined classification.

2.1 Initialization. $i = 0$, $S_{\text{del}} = \Phi$. Determine S_{add} according to the current separating hyperplane. Specify $C_{\text{tmp}}^*(i) = 10^{-3} * C$.

2.2 Progressive Classification.

do

2.2.1 Update S_{train} and S_{test}.

$S_{\text{train}} = S_{\text{train}} - S_{\text{del}} + S_{\text{add}}$,

$S_{\text{test}} = S_{\text{test}} - S_{\text{add}} + S_{\text{del}}$.

2.2.2 Execute the *tSVM* on S_{train} and S_{test} based on $C_{\text{tmp}}^*(i)$.

2.2.3 Determine S_{add} and S_{del}.

if size$(S_{\text{add}}^{+}) > 0$ and size$(S_{\text{add}}^{-}) > 0$

$S_{\text{add}} = S_{\text{add}}^{+} + S_{\text{add}}^{-}$.

else if size$(S_{\text{add}}^{+}) > 0$ and size$(S_{\text{add}}^{-}) = 0$

$S_{\text{add}} = S_{\text{add}}^{+}$.

else if size$(S_{\text{add}}^{+}) = 0$ and size$(S_{\text{add}}^{-}) > 0$

$S_{\text{add}} = S_{\text{add}}^{-}$.

else

$S_{\text{add}} = \Phi$.

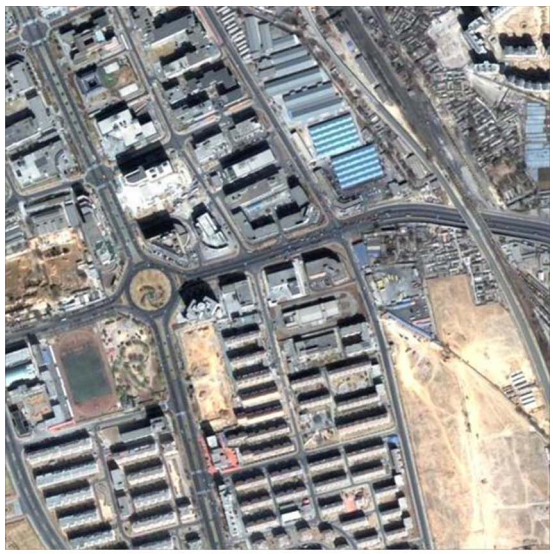
end

$S_{\text{del}} = \{\mathbf{x}_j | f(\mathbf{x}_j) \neq y_j\}$, where $\mathbf{x}_j \in S_{\text{train}}$ and y_j is achieved at ith iteration.

2.2.4 $i = i+1$, $C_{\text{tmp}}^*(i) = \min(2 \times C_{\text{tmp}}^*(i-1), C^*)$.

while ($S_{\text{add}} \neq \Phi$)

3. Achieve the final change map by applying the tSVM on S_{train} and S_{test}.



(a)

(b)

Fig. 2. Images used in this letter. (a) Image in 2002. (b) Image in 2003.

III. Experiments and Discussion

To assess the effectiveness of the proposed approach, experiments were conducted on several different sets of VHR images. For space limitation, only the results on one data set are shown in this letter. The images are taken over Beijing (China), acquired by QuickBird on April 9, 2002 (at nadir), and November 12, 2003 (23° off-nadir look angle). The images consist of panchromatic images (61 cm/pixel) and multispectral images (2.4 m/pixel, R/G/B/Nir-bands). To achieve high spatial resolution and high spectral resolution simultaneously, the panchromatic images and the multispectral images are merged by Ehlers fusion [9] in the preprocessing step. The sharpened pseudocolor images of the test sites are shown in Fig. 2.

A. Experiments Description

For object-level change detection, the segmentation algorithm is a key issue since it affects object-specific change feature extraction and object-specific change feature classification.

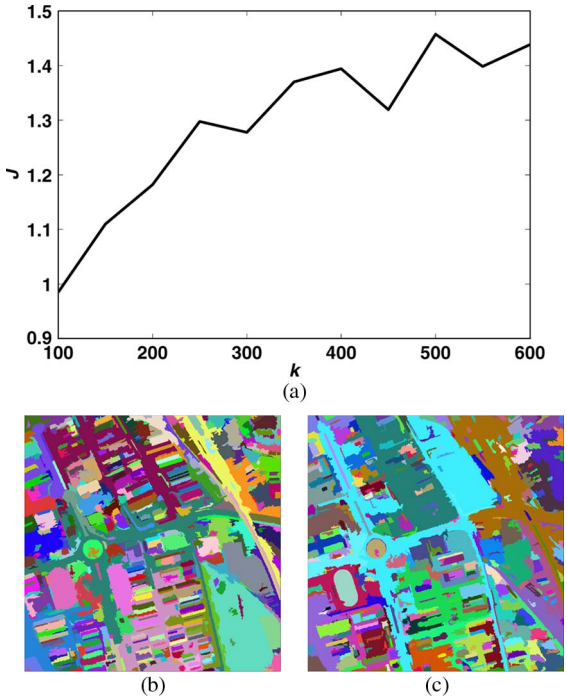


Fig. 3. Results of segmentation and multitemporal segmentation. (a) Segmentation performance versus k. (b) Segmentation result of Fig. 2(a), $k = 500$. (c) Multitemporal segmentation, $k = 500$.

As for the segmentation method used in this letter, three parameters need to be selected: σ, m, and k. σ is used to smooth the input image before segmentation, and m is the minimum component size enforced by postprocessing. On the whole, the performance is relatively stable when changing these two parameters. In this letter, $\sigma = 0.5$, and $m = 200$. k is the value for the threshold function and controls the region size, and the segmentation result suffers somewhat from sensitivity to k. Since multitemporal segmentation is to maximize the distance between different regions and to minimize the difference within the homogeneous region, we use J (6) to measure the segmentation performance, where $\mathbf{S}_b = \sum_{i=1}^{N}(\mathbf{m}_i - \mathbf{m})^T$ denotes the interclass distance, $\mathbf{S}_w = \sum_{i=1}^{N}(1/N_i)\sum_{k=1}^{N_i}(\mathbf{d}_k^{(i)} - \mathbf{m}_i)^T$ denotes the intraclass distance, $\mathbf{d}_k^{(i)}$ is the C-dimension spectral vector, and $\mathbf{m}_i$ is the mean spectral vector within the ith region. Fig. 3(a) shows the changes in J with k. The result is relatively stable to k once the image resolution is given, and the best result is achieved when $k = 500$. Fig. 3(b) and (c) shows the correspondent segmentation result of Fig. 2(a) and the multitemporal segmentation result, respectively. By this way, k can be determined automatically.

$$J = \text{trace}\left(\frac{\mathbf{S}_b}{\mathbf{S}_w}\right). \tag{6}$$

The classification performance is affected by the SVM model and the regularization parameter C. As with most other researchers' work, we use a Gaussian radial basis function kernel $\mathbf{K}(\mathbf{x}, \mathbf{x}_i) = \exp(-\|\mathbf{x} - \mathbf{x}_i\|^2/2\sigma^2)$. In the *iSVM*, coarse- to fine-grid search [10] and fivefold cross-validation are used for model selection by setting $C \in [0.01, 500]$ and $\sigma^2 \in [0.1, 1]$. Before the initial classification, α, the ratio of the training samples, needs to be determined. As illustrated from Fig. 4, both models can achieve good performances when α lies between 0.25 and 0.32. In the following experiments, we set $\alpha = 0.3$.

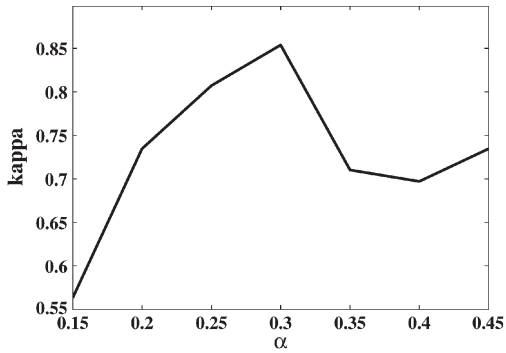


Fig. 4. Plot of the changing ratios for different $kappa$ coefficients.

TABLE I
PERFORMANCE COMPARISON AGAINST DIFFERENT APPROACHES

accuracy	*threshold*	*rSVM*	*itSVM*	*tSVM*	*iSVM*
MAs	11287	18029	22259	9677	25981
FAs	167336	144228	47501	142718	76501
OAs	178623	162257	69760	152395	102482
kappa	0.66	0.68	0.85	0.70	0.79

B. Experimental Results

Two sets of experiments are designed to validate the effectiveness of the proposed approach. The first set of experiments is to demonstrate the importance of the original multitemporal signatures in improving the discriminability between the changed class and the unchanged class. The second set of experiments aims to analyze the advantages of progressive classification. We use missed alarms (MAs), false alarms (FAs), overall alarms (OAs), and $kappa$ (kappa coefficient) [11] to evaluate the performances of the different approaches.

In the first set of experiments, we compare *itSVM* (the proposed approach) with *threshold* (classifying the object-specific relative change features with the optimal threshold, the optimal threshold is selected by a try-and-error strategy) and *rSVM* (classifying the relative change features with the progressive *tSVM*). The correspondent performances are listed in Table I. First, *threshold* is not comparable with *itSVM* since the relative change features are not linearly separable. Although using the same classification strategy with *itSVM*, *rSVM* cannot achieve a good performance since only the relative change features are not discriminative enough to represent the complex changes. This comparison demonstrates the necessity of the original multitemporal signatures.

To demonstrate the advantages of the progressive change feature classification, in the second set of experiments, the proposed approach is compared against different classification strategies, including *iSVM* and *tSVM*. Fig. 5(a) shows the reference change mask, and Fig. 5(b)–(d) shows the results by the different approaches, from which we can see that most changes are detected correctly by either the *iSVM* or the *tSVM*. This validates the effectiveness of the combination of the relative change features and the original multitemporal signatures and

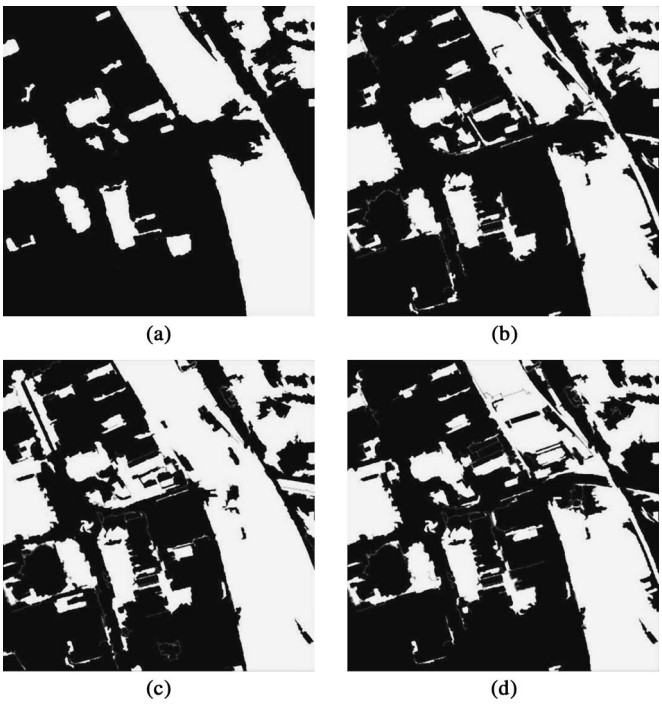


Fig. 5. Result comparison of different methods. (a) Reference change mask. (b) Change mask by *itSVM*. (c) Change mask by *tSVM*. (d) Change mask by *iSVM*.

the effectiveness of the SVM in classifying the change features. In detail, the proposed approach is superior to the other two procedures, $kappa$ increases from 0.70 by *tSVM* and 0.79 by *iSVM* to 0.85 by *itSVM*, and the OAs decreases from 152 395 by *tSVM* and 102 482 by *iSVM* to 69 760 by *itSVM*. The improvements lie in the following facts. On the one hand, *tSVM* combines the unlabeled data in constructing the separating hyperplane; however, influenced by some disturbed initial training samples which may have incorrect labels, many unchanged regions are detected as wrongly changed. Moreover, this influence is so strong that FAs are decreased significantly even by *iSVM*. On the other hand, *itSVM* outperforms the aforementioned two approaches by the combination of iterative classification and transductive classification, iterative classification reduces the OAs by gradually removing the false training samples and dynamically adding new representative training samples, while transductive classification improves the performance by considering the unlabeled data in investigation and tuning the separating hyperplane based on the iterative rectification of the training samples. In other words, not the transductive classification but the combination of transductive classification and iterative classification helps improve the performance.

One distinctive feature of the proposed approach is the high efficiency. As for a 1024×1024 image pair, it will take several hours for the SVM to classify more than 1 000 000 change features by the pixel-based approach, while the proposed approach can finish the whole task in less than 2 min on a general personal computer. The classification time is determined by the number and the discriminability of the change features. The number of regions extracted by multitemporal segmentation is much less than the number of pixels, and the discriminability between the changed class and the unchanged class is improved by the object-specific change features. Due to the aforementioned factors, the proposed approach is much faster than the traditional pixel-by-pixel classification.

IV. Conclusion

A novel approach has been presented for change detection of VHR images; the novelties lie in the following three aspects: 1) Fast multitemporal segmentation and object-specific change feature classification make the proposed approach very fast; 2) object-specific change feature representation and the *tSVM* make the proposed approach high in accuracy; and 3) the iterative *tSVM* makes the proposed approach automated.

Despite of the promising preliminary results, many developments need to be considered in the future work. For example, since the initial training samples are selected based on the spectral difference, the proposed approach may fail when the spectral changes of the unchanged regions are more complex. More discriminative features and more robust difference measures should be considered for the object-specific change feature representation. Future developments are also related to more effective model selection technique such as similarity-measure-based procedure [12].

Acknowledgment

The authors would like to thank P. F. Felzenszwalb and F. Sinz for providing the related source code and the anonymous reviewers for the suggestions to improve this letter.

References

[1] P. Aplin, P. M. Atkinson, and P. J. Curran, "Fine spatial resolution satellite sensors for the next decade," *Int. J. Remote Sens.*, vol. 18, no. 18, pp. 3873–3881, Dec. 1997.

[2] T. Blaschke and J. Strobl, "What's wrong with pixels? Some recent developments interfacing remote sensing and GIS," *GIS Zeitschrift Geoin.*, vol. 6, pp. 12–17, 2001.

[3] X. Dai and S. Khorram, "Requirements and techniques for an automated change detection system," in *Proc. IGARSS*, 1998, pp. 2752–2754.

[4] P. F. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, Feb. 2004.

[5] R. Unnikrishnan, C. Pantofaru, and M. Hebert, "Toward objective evaluation of image segmentation algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 929–944, Jun. 2007.

[6] F. Bovolo and L. Bruzzone, "A theoretical framework for unsupervised change detection based on change vector analysis in the polar domain," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 1, pp. 218–236, Jan. 2007.

[7] R. Collobert, F. Sinz, J. Weston, and L. Bottou, "Large scale transductive SVMs," *J. Mach. Learn. Res.*, vol. 7, pp. 1687–1712, 2006.

[8] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, pp. 409–415.

[9] M. Ehlers, "Spectral characteristics preserving image fusion based on Fourier domain filtering," *Proc. SPIE*, vol. 5574, pp. 1–14, 2004.

[10] H. Li, S. Wang, and F. Qi, "SVM model selection with the VC bound," in *Proc. CIS*, vol. 3314, *Lecture Notes in Computer Science*, 2004, pp. 1067–1071.

[11] R. G. Congalton and K. Green, *Assessing the Accuracy of Remotely Sensed Data: Principles and Practices*., 2nd ed. Boca Raton, FL: CRC Press, 1998.

[12] F. Bovolo, L. Bruzzone, and M. Marconcini, "A novel approach to unsupervised change detection based on a semisupervised SVM and a similarity measure," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 7, pp. 2070–2082, Jul. 2008.