ISPRS Journal of Photogrammetry and Remote Sensing 185 (2022) 16–31

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

Structured graph based image regression for unsupervised multimodal change detection

Yuli Sun [a], Lin Lei [a,*], Xiang Tan [a], Dongdong Guan [b], Junzheng Wu [a], Gangyao Kuang [a]

[a] College of Electronic Science, National University of Defense Technology, Changsha 410073, China
[b] High-Tech Institute of Xi'an, Xi'an 710025, China

ARTICLE INFO

Keywords:
Unsupervised change detection
Structured graph
Hypergraph
Image regression
Multimodal
Markov random field

ABSTRACT

Change detection for multimodal remote sensing images is an important and challenging research topic with a wide range of applications in disaster assessment and environmental monitoring. To address the problem that heterogeneous images cannot be directly compared due to different imaging mechanisms, we propose an unsupervised image regression method based on the inherent structure consistency between heterogeneous images, which learns a structured graph and computes the regression image by graph projection. Firstly, the proposed method uses the self-expression property to preserve the global structure of image and uses the adaptive neighbor approach to capture the local structure of image in the graph learning process. Then, with the learned graph, two types of structure constraints are introduced into the regression model: one corresponds to the global self-expression constraint and the other corresponds to the local similarity constraint, which can be further implemented by using graph or hypergraph Laplacian based regularization. Finally, a Markov segmentation model is designed to calculate the binary change map, which combines the change information and spatial information to improve the detection accuracy. Experiments conducted on six real data sets show the effectiveness of the proposed method by comparing with five state-of-the-art algorithms, achieving 2.4%, 5.5% and 4.1% improvements in accuracy, Kappa coefficient, and F1 score respectively. Source code of the proposed method will be made available at https://github.com/yulisun/GIR-MRF.

1. Introduction

1.1. Background

Multimodal or heterogeneous change detection (CD) (Mercier et al., 2008) is an increasingly active and challenging research topic in remote sensing for Earth observation, which aims to identify changes that happened on the Earth by comparing two satellite images acquired at different times over the same geographical area, but under heterogeneous conditions. Multimodal CD (MCD) can be regarded as a generalization of the basic monomodal or homogeneous CD problem (Li et al., 2020).

MCD can exploit the huge amount of remote sensing data to extract reliable information about the land cover changes as it can relax the restriction of homogeneous conditions. The input images could be acquired by different sensors (e.g., a multispectral image at t_1 and a synthetic aperture radar (SAR) image at t_2), or recorded with different sensor parameters (e.g., a near-infrared (NIR) band image at t_1 and a RGB bands image at t_2, or a C-band SAR image at t_1 and an X-band SAR image at t_2), or under dramatically different environmental conditions (weather or light) that comparisons cannot be made except through meticulous preprocessing and co-calibration (Mignotte, 2020). MCD has advantages in two areas: first, it can increase the temporal resolution or extend the time frame of long-term trend monitoring by inserting heterogeneous data (Chen et al., 2019); second, it is particularly useful to shorten the response time of CD, which has a very urgent need in rescue and assessment of emergency disasters, such as flood and earthquake. MCD allows to use the first image of opportunity to detect changes instead of waiting for a comparable homogeneous image to be acquired, what's more, due to the accompanying adverse light and weather conditions, the acquired homogeneous image may not be available (Brunner et al., 2010; Saha et al., 2021a; Ebel et al., 2021).

MCD is a challenging task since the multitemporal images cannot be directly compared to obtain the difference image (DI) as in homogeneous CD. Therefore, the core task of MCD is to make the heterogeneous images comparable. Let $\mathbf{X}$ and $\mathbf{Y}$ be two co-registered images to be

* Corresponding author.
E-mail address: leilin19@sina.com (L. Lei).

https://doi.org/10.1016/j.isprsjprs.2022.01.004
Received 6 July 2021; Received in revised form 6 January 2022; Accepted 6 January 2022

compared, and let x and y be two data samples drawn on the same spatial location from $\mathbf{X}$ and $\mathbf{Y}$, respectively. Depending on the basic analysis unit of CD method, x and y can be individual pixels, square patches or superpixels. Let H_0 and H_1 indicate the "unchanged" and "changed" hypotheses, respectively. Then in the homogeneous CD, we can directly compare x and y with arithmetical operators such as difference operator (Bovolo et al., 2011) or ratio/log-ratio operator (Moser and Serpico, 2006; Zhang et al., 2021) (the former is usually used in optical images and the latter is usually used in SAR images), which is based on the assumption that x and y exhibit the same (or similar) distribution when conditioned to H_0. However, this assumption is violated in MCD as heterogeneous images provide different descriptions of the same object and exhibit quite different characteristics.

Generally, most MCD methods are broadly to transform the "incomparable" images to a new common domain, or to transform one image to the domain of the other image, such that they are "comparable" (Moser et al., 2020). For the former transformation, we can express it as two mappings of $\phi : x \to w$ and $\varphi : y \to w'$, such that w and w' have the same (or very similar) distribution given H_0. According to the transformed domain (w and w'), the transform-based MCD methods can be roughly divided into: 1) classification-based methods, which transform the multimodal images to a common category space, such as the post-classification comparison (PCC) (Jensen et al., 1987), the multi-temporal segmentation and compound classification (MS-CC) (Wan et al., 2019), and classified adversarial network (CAN) (Wu et al., 2020). 2) Feature space-based methods, which transform the multimodal images to a common constructed or learned feature space, such as methods based on manually constructed features: sorted histogram method (SH) (Wan et al., 2018), manifold learning based method (Prendes et al., 2014; Prendes et al., 2016), pixel pairwise-based methods (Touati et al., 2019a, 2019b; Touati and Mignotte, 2017), kernel canonical correlation analysis (kCCA)-based method (Volpi et al., 2015), and methods based on learned deep features: the symmetric convolutional coupling network (SCCN) (Liu et al., 2018), logarithmic transformation feature learning network (LT-FL) (Zhan et al., 2018), deep feature representation and mapping transformation based method (DFR-MT) (Zhang et al., 2016), and anomaly feature learning-based deep sparse residual model (AFL-DSR) (Touati et al., 2020).

For the latter transformation, it can be regarded as image regression or image translation and expressed as $\xi : x \to y'$ or $\varsigma : y \to x'$, such that y' and y, or x' and x, have the same distribution when conditioned to H_0. According to the mapping ξ or ς, the image regression based MCD methods can be roughly divided into: 1) traditional signal-processing methods. Homogeneous pixel transformation (HPT) uses kernel regression on a sample of k-nearest neighbor (KNN) pixels to construct mappings (ξ or ς) between the input images (Liu et al., 2017). The mappings are learned with the labeled unchanged training data. To avoid the reliance on labeled data, an unsupervised affinity matrix difference based image regression (AMD-IR) is proposed in Luppino et al. (2019), which first uses AMD to identify pixels that are likely to be unchanged as pseudo-training data, and then completes the image translation by four regression methods: Gaussian process regression, support vector regression, random forest regression, and the HPT. 2) Deep learning methods. In Niu et al. (2018), a conditional generative adversarial network (GAN) is used to translate the optical image to the SAR image and an approximate network is built to approximate the original SAR image to the translated one. A coupling translation networks (CPTN) is proposed in Gong et al. (2019), which uses two variational autoencoders (VAE) to extract a shared-latent space for heterogeneous images, and then uses a coupled GAN to translate the heterogeneous images into each other's domain from their shared-latent space. Based on the AMD change prior, two deep image translation methods are proposed in Luppino et al. (2021): the X-Net composed of two fully convolutional networks and the adversarial cyclic encoder network (ACE-Net) composed of two autoencoders whose code spaces are aligned by adversarial training. In Jiang et al. (2020), a deep homogeneous feature fusion (DHFF) is proposed to transform the SAR image to the optical image domain, which is based on an iterative image style transfer strategy.

Note that the mappings of (ξ, ς) usually need to be trained with unchanged pairs of heterogeneous data. Therefore, these regression based MCD methods either require a labeled training set under the supervision mode (such as HPT), or need a pre-constructed pseudo-training set/change prior to guide the training process (such as AMD-IR, X-net and ACE-Net), or involve a complex iterative coarse-to-fine filtering process to construct the pseudo-training set while using it to learn the mappings (ξ, ς) (such as conditional GAN, CPTN and DHFF).

1.2. Motivation

In this paper, we aim to propose an unsupervised image regression method for MCD, which does not need any labeled data. Specifically, the proposed method is based on the self-similarity property of images, that is, any small part of the image can always find similar parts within the same image, which has been widely used in the image denoising with the so-called "nonlocal-based" methods (Dabov et al., 2007; Guan et al., 2018), and has also recently been explored by researchers for MCD (Lei et al., 2020; Mignotte, 2020; Sun et al., 2020, 2021a, 2021b). In Lei et al., 2020; Sun et al., 2021a, 2021b, the self-similarity is used to construct graphs (fixed KNN graph or adaptive probabilistic graph) to represent the structure of each image and can be regarded as $\phi : x \to w$, $\varphi : y \to w'$ with w and w' denoting graphs, and then the graphs (w and w') are compared to calculate the DI by graph projection. In Mignotte (2020) and Sun et al. (2020), the self-similarity is used to project the pre-event image to the domain of post-event image as $\xi : x \xrightarrow{w} y'$ by fractal projection with fractal code w and image reconstruction with a self-expression matrix w respectively, where w can represent the similarity relationships between image patches (w can also be defined as the structure of image). In these self-similarity based MCD methods, two points are especially important: first, how to construct the structure w of image; and second, how to measure the structure difference to detect the changes. Hence, we propose a structured graph based image regression method to improve the MCD performance from the above two points.

First, a main challenge in the unsupervised MCD is that transformations must be learnt from a dataset that includes noise and changes, which means that the noise and changed pixels will contaminate the mappings of (ϕ, φ) and (ξ, ς) and make them less effective. To address this challenge, the proposed method uses the inherent imaging-modality-invariant structure consistency between heterogeneous images to compute the regression image. It is more robust to noise than the common regression-based methods that aim to learn a luminance transformation function (ξ or ς) to translate the image. What's more, to alleviate the negative effects of changed pixels, the proposed method decomposes the post-event image into the regression image of pre-event and the changed image in the regression process. At the same time, the prior sparse knowledge that only a small part of image changed in the event is also taken into account within the regression model.

Second, local and global structure information are adaptively combined to construct a graph to represent the structure of image. On the one hand, it learns a probabilistic graph by automatically selecting the most informative neighbors for each node and adaptively assigning weights instead of constructing KNN graph manually as in Sun et al., 2021a, 2021b or ε-nearest-neighbor (ε NN) graph, which relies heavily on the choice of k and ε. Therefore, the learned graph in the proposed method is more robust and not sensitive to noise and outliers. On the other hand, it can preserve both local and global structure information by combining local similarity regularization and global self-expression in the graph learning framework. As demonstrated in many problems, such as feature selection (Zhu et al., 2016), classification (Zhou et al., 2004), clustering (Wang et al., 2009; Kang et al., 2021), local and global

structure information are both important to graph performance since they can provide complementary information to each other. Therefore, the learned graph can contain more information and is more representative than graph containing only local structure information as in Sun et al., 2021a, 2021b and the graph containing only global information as in Sun et al. (2020).

Third, to improve the regression performance, two types of constraints are introduced into the regression model: one corresponds to the global self-expression constraint and the other corresponds to the local similarity constraint, which are relative to the graph learning process. Moreover, in order to preserve the local structure information to improve the regression performance, two strategies are used in the proposed method: graph Laplacian based regularization (GLR) and hypergraph Laplacian based regularization (HGLR). HGLR makes the vertices with the similar characteristics be enclosed by a hyperedge, which is capable to connect more than two vertices (Zhou et al., 2006; Wang et al., 2020) and thus captures the high-order information. Both GLR and HGLR enhance the robustness of image regression.

Fourth, since the proposed method uses a graph-based approach to complete the image regression and calculate the DI, naturally, we use the graph cut method to obtain the binary changed map (CM) based on a Markov random field (MRF) segmentation model, which uses Gaussian mixture model to construct the change energy term and combines the spatial context information of DI and similarity information of the original images to construct the spatial energy term.

1.3. Contribution

In particular, our contribution refers to a graph based unsupervised image regression method for MCD.

- An inherent structure consistency based method is proposed for MCD, which translates the pre-event image to the domain of post-event image by using a graph-based regression model.
- A robust graph is adaptively constructed to capture both local and global structural information of the image.
- The global self-expression constraint and local similarity based GLR or HGLR are used to improve the regression performance.
- A MRF segmentation model is designed to combine the change information and spatial information, which is solved by the graph cut.

1.4. Outline and notation

The rest of this paper is structured as follows: Section 2 describes the related graph learning techniques and basic concepts of hypergraph. Section III describes the details of the proposed MCD method. Section 4 presents the experimental results by comparing with some existing state-of-the-art (SOTA) methods and gives some discussions. Finally, we conclude this paper in Section 5. For convenience, Table 1 lists some important notations used in the rest of this paper.

2. Preliminaries

Since the proposed method in this paper is to use the structured graph for MCD, we briefly review local and global structure learning, and then introduce some basic concepts of hypergraph in this section.

2.1. Local structure learning

It is intuitive that the similarity $w_{i,j}$ between the i-th sample $\mathbf{X}_i$ and the j-th sample $\mathbf{X}_j$ is larger if the distance between $\mathbf{X}_i$ and $\mathbf{X}_j$ is smaller. Two types of local structure based graph have been used in MCD (Lei et al., 2020; Sun et al., 2021a, 2021b). The first one is the widely used KNN graph, which computes the similarity matrix $\mathbf{W}$ as

Table 1
List of important notations.

Symbol	Description
$\widetilde{\mathbf{X}}, \widetilde{\mathbf{Y}}, \widetilde{\mathbf{Z}}$	pre-event, post-even, and regression images
$\mathbf{X}, \mathbf{Y}, \mathbf{Z}$	feature matrices of $\widetilde{\mathbf{X}}, \widetilde{\mathbf{Y}}, \widetilde{\mathbf{Z}}$
$\mathbf{X}_i$	i-th column of a matrix $\mathbf{X}$
$x_{i,j}$	i-th row and j-th column element of $\mathbf{X}$
$\mathbf{X}^{(t)}$	the t-th iteration of $\mathbf{X}$
$\|\mathbf{X}\|_F$	Frobenius norm of $\mathbf{X}$, $\|\mathbf{X}\|_F = \sqrt{\sum_i \sum_j x_{ij}^2}$
$\|\mathbf{X}\|_1$	ℓ_1-norm of $\mathbf{X}$, $\|\mathbf{X}\|_1 = \sum_i \sum_j \|x_{ij}\|$
$\|\mathbf{X}\|_{2,1}$	$\ell_{2,1}$-norm of $\mathbf{X}$, $\|\mathbf{X}\|_{2,1} = \sum_j \sqrt{\sum_i x_{ij}^2}$
$Tr(\mathbf{X})$	trace of $\mathbf{X}$
$G = (V, E, w)$	graph with vertex set V, edge set E and weight w
$G^h = (V^h, E^h, w^h)$	hypergraph with vertex set V^h, hyperedge E^h and hyperedge weight w^h
$\mathbf{D}_v$	vertex degree matrix of hypergraph G^h
$\mathbf{D}_e$	hyperedge degree matrix of hypergraph G^h
$\mathbf{W}^h$	hyperedge weight matrix of hypergraph G^h
$\mathbf{L}^g, \mathbf{L}^h$	graph and hypergraph Laplacian matrix
$\mathbf{I}_N$	an $N \times N$ identity matrix
$\mathbf{1}_N$	an $N \times 1$ column vector of ones
$\mathbf{X} \geqslant 0$	nonnegative matrix

$$w_{i,j} = \begin{cases} \exp\left(-\eta dist_{i,j}^x\right), & \mathbf{X}_i \text{ is one of the KNN of } \mathbf{X}_j \\ 0, & \text{otherwise} \end{cases}, \tag{1}$$

where $dist_{i,j}^x$ represents the distance between $\mathbf{X}_i$ and $\mathbf{X}_j$, such as $dist_{i,j}^x = \left\|\mathbf{X}_i - \mathbf{X}_j\right\|_2^2$, and $\eta > 0$ is a tuning parameter. Although KNN is very intuitive, it is heavily dependent on the choice of parameters (k and η), which is hard to set with a general strategy.

The second one is the adaptive neighbor approach, which is first proposed for classification in Nie et al. (2014). This approach learns the similarity matrix $\mathbf{W}$ by solving the following objective function

$$\min_{\mathbf{W}_j} \sum_{i=1}^{N} dist_{i,j}^x w_{i,j} + \alpha w_{i,j}^2 \quad s.t. \quad 0 \leqslant w_{i,j} \leqslant 1, \ \sum_{i=1}^{N} w_{i,j} = 1, \tag{2}$$

where $\alpha > 0$ is a tuning parameter. As we will show latter, the matrix $\mathbf{W}$ can be column k-sparse when appropriate α is chosen. $\mathbf{W}$ also describes the relationship between sample and its nearest neighbors.

2.2. Global structure learning

Self-expression property has been applied to many applications for its ability in capturing the global structure of data (Huang et al., 2019; Zhang et al., 2018), which assumes that each sample can be represented as a linear combination of other samples, i.e., $\mathbf{X}_i \approx \sum_{j=1}^{N} \mathbf{X}_j w_{j,i}$. The coefficient matrix $\mathbf{W}$ can be regarded as the similarity matrix. Rather than focusing on the predefined local neighborhood (such as KNN), we can learn the matrix $\mathbf{W}$ by solving the following minimization problem

$$\min_{\mathbf{W}} \|\mathbf{X} - \mathbf{X}\mathbf{W}\|_F^2 + \alpha f(\mathbf{W}), \tag{3}$$

where $f(\mathbf{W})$ is a regularizer on $\mathbf{W}$ and $\alpha > 0$ is a tuning parameter. As (3) uses all the samples to reconstruct the target sample and learns $\mathbf{W}$ automatically, it is supposed to capture the global structure information of $\mathbf{X}$.

2.3. Basic concepts of hypergraph

In contrast to the pair-wise graph, a hypergraph can link more than

two vertices. Denote V^h, E^h as the vertex set and hyperedge set corresponding to a hypergraph of $G^h = \{V^h, E^h, w^h\}$, respectively. Each hyperedge e is a subset of the vertex set V^h. Denote the weight associate with the hyperedge $e \in E^h$ as $w^h(e)$. The degree of a vertex $v \in V^h$ is defined as $d(v) = \sum_{\{e \in E^h | v \in e\}} w^h(e)$, and the degree of a hyperedge e is defined as $\delta(e) = |e|$. Denote the incident matrix $\mathbf{H}$ as a $|V^h| \times |E^h|$ matrix, whose entry satisfies $h(v,e) = 1$ if $v \in e$, and $h(v,e) = 0$ otherwise. With these definitions, we have

$$d(v) = \sum_{e \in E^h} w^h(e) h(v,e), \quad \delta(e) = \sum_{v \in V^h} h(v,e). \tag{4}$$

Let $\mathbf{D}_v$ and $\mathbf{D}_e$ be the diagonal matrices containing the degree of each vertex and hyperedge, respectively, and denote $\mathbf{W}^h$ as the diagonal matrix of the edge weight. Then, the unnormalized hypergraph Laplacian matrix is defined as Zhou et al. (2006), Agarwal et al. (2006)

$$\mathbf{L}^h = \mathbf{D}_v - \mathbf{H}\mathbf{W}^h\mathbf{D}_e^{-1}\mathbf{H}^T. \tag{5}$$

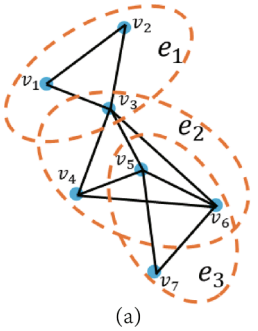
Fig. 1 gives an example of a hypergraph.

3. Methodology

We consider a pair of co-registered heterogeneous images acquired at time t_1 (pre-event) and t_2 (post-event), which are denoted as $\widetilde{\mathbf{X}} \in \mathbb{R}^{M \times N \times C_{\mathbf{X}}}$ in domain $\mathscr{X}$ and $\widetilde{\mathbf{Y}} \in \mathbb{R}^{M \times N \times C_{\mathbf{Y}}}$ in domain $\mathscr{Y}$, respectively. We define their pixels as $\widetilde{x}(m,n,c)$ and $\widetilde{y}(m,n,c)$, respectively. Here, M, N and $C_{\mathbf{X}}$ ($C_{\mathbf{Y}}$) define the height, width, and the number of bands of the image $\widetilde{\mathbf{X}}$ ($\widetilde{\mathbf{Y}}$), respectively.

As illustrated in the introduction, we cannot apply the traditional arithmetical operators (such as difference and ratio/log-ratio operators) to calculate the DI in MCD, since it is meaningless to directly compare entities from different domains. The strategy is instead by measuring the structure consistency between heterogeneous images. As illustrated in Fig. 2, each image is divided into small parts with the same segmentation form. For the pre-event image, with the inherent self-similarity property, each small part of the image can always find some similar parts within the same image. That is, if $\widetilde{\mathbf{X}}_i$ and $\widetilde{\mathbf{X}}_j$ represents the same kind of object and showing that they are very similar, and neither of them changes during the event, then $\widetilde{\mathbf{Y}}_i$ and $\widetilde{\mathbf{Y}}_j$ also represents the same kind of object in the post-event image and showing that they are also very similar. Furthermore, if $\widetilde{\mathbf{X}}_i$ can be represented by these similar $\widetilde{\mathbf{X}}_j$, then $\widetilde{\mathbf{Y}}_i$ can also be represented by these $\widetilde{\mathbf{Y}}_j$, as long as they have not changed. We use the similarity relationships between $\widetilde{\mathbf{X}}_i$ and $\widetilde{\mathbf{X}}_j$ to represent the structure of $\widetilde{\mathbf{X}}_i$. Then, we can find that the structure of $\widetilde{\mathbf{X}}_i$ can be well conformed by the unchanged $\widetilde{\mathbf{Y}}_i$, as shown in the unchanged part of Fig. 2. On the contrary, if $\widetilde{\mathbf{Y}}_i$ has changed in the event, the structure of $\widetilde{\mathbf{X}}_i$ is no longer preserved by $\widetilde{\mathbf{Y}}_i$, showing that the similarity relationships between $\widetilde{\mathbf{Y}}_i$ and $\widetilde{\mathbf{Y}}_j$ are quite different from that of $\widetilde{\mathbf{X}}_i$ and $\widetilde{\mathbf{X}}_j$, as shown in the changed part of Fig. 2. Therefore, we can transform the pre-event image to the domain of post-event image by using the structure consistency, which is quite imaging modality invariant, and then measure the change level by comparing the regression image and the post-event image.

There are three main problems to be solved: how to represent the image structure, how to complete the image regression to obtain the DI, and how to compute the final CM. Fig. 3 shows the framework of the proposed method, which consists of four steps: 1) preprocessing; 2) structure representation by constructing graph; 3) image regression by using structure consistency; 4) binary CM calculation with MRF segmentation.



(a)

	e_1	e_2	e_3
v_1	1	0	0
v_2	1	0	0
v_3	1	1	0
v_4	0	1	0
v_5	0	1	1
v_6	0	1	1
v_7	0	0	1

(b)

Fig. 1. An example of a hypergraph: (a) A hypergraph represents the complex relationships of seven points; (b) Incidence matrix $\mathbf{H}$ of the hypergraph, whose entry $h(v_i, e_j) = 1$ if a hyperedge e_j contains the i-th vertex v_i and $h(v_i, e_j) = 0$ otherwise.

3.1. Preprocessing

Preprocessing consists of two main tasks: superpixel segmentation and feature extraction. In the proposed method, it chooses the image block (superpixel) that represents the same kind of object as the basic analysis unit rather than the individual pixel or square image patch. This brings two advantages: first, it can maintain the shape and edge of object, and contain the context information because each superpixel internally belongs to the same kind of object; second, it can reduce the size of graph, thus reducing the computational complexity, which is very useful for MCD of very-high-resolution images.

We use the simple linear iterative clustering (SLIC) method (Achanta et al., 2012) to generate the superpixels for its superior in both efficiency (linear complexity of the number of image pixels) and boundary preservation. SLIC is easy to use, offers flexibility in the compactness and number of the superpixels it generates, and is straightforward to extend to higher dimensions. We first apply SLIC on each image to generate the superpixels. For different types of image (e.g. $\widetilde{\mathbf{X}}$), we use different adjustments on the SLIC. For the optical image $\widetilde{\mathbf{X}}$ with RGB bands, the original SLIC is directly employed to generate the superpixels; for the multispectral image $\widetilde{\mathbf{X}}$ with $C_{\mathbf{X}} > 3$, the principle component analysis (PCA) method is used to obtain the first three principle components, and then SLIC is used to segment the image; for the SAR image $\widetilde{\mathbf{X}}$, since it is usually assumed to be contaminated by the multiplicative speckle noise with Gamma distribution, the Euclidean distance in original SLIC is not proper to generate superpixels. Inspired by several similarity criteria proposed in Deledalle et al. (2012), we use the following pixel intensity distances to replace the Euclidean distance in SLIC

$$d = \log\left(\frac{\widetilde{x}_i + \widetilde{x}_j}{2\sqrt{\widetilde{x}_i \widetilde{x}_j}}\right) \quad \text{or} \quad d = \left(\log(\widetilde{x}_i) - \log(\widetilde{x}_j)\right)^2, \tag{6}$$

where $\widetilde{x}_i$ and $\widetilde{x}_j$ are intensity values of two pixels of SAR image.

Once the SLIC (or adjusted SLIC) is performed on each image independently, we can obtain the segmentation map of each image, denoted as Λ^x and Λ^y. By taking the intersection of segmentation maps Λ^x and Λ^y, eliminating the empty sets and merging the very small regions into the nearest regions, we can obtain the co-segmentation map $\Lambda = \{\Lambda_i | i = 1, \cdots, N_S\}$ and the segmented superpixels of $\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{Y}}$ (Touati et al., 2019b), denoted as $\widetilde{\mathbf{X}}_i = \{\widetilde{x}(m,n,c) | (m,n) \in \Lambda_i, c = 1, \cdots, C_{\mathbf{X}}\}$ and $\widetilde{\mathbf{Y}}_i = \{\widetilde{y}(m,n,c) | (m,n) \in \Lambda_i, c = 1, \cdots, C_{\mathbf{Y}}\}$ respectively with $i = 1, \cdots, N_S$. Since the co-segmentation map Λ is an intersection of Λ^x and Λ^y, the set of pixels inside each superpixel in Λ have the property to be internally homogeneous simultaneously in $\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{Y}}$. There are other superpixel segmentation methods that can also be used in the preprocessing, such as the multitemporal segmentation strategies in Bovolo (2009), Saha et al. (2021b).

With the segmentation map Λ, different kinds of feature information

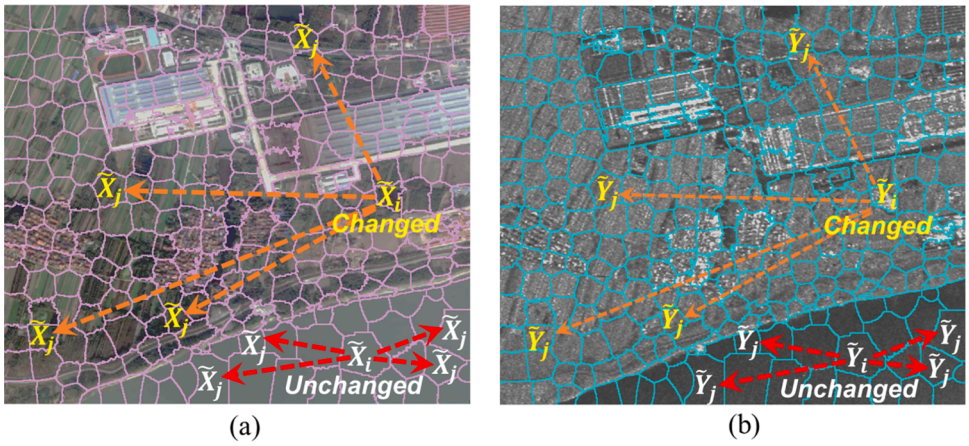


Fig. 2. Illustration of the structure consistency in heterogeneous images: (a) optical image; (b) SAR image. The similarity between image parts is reflected by the thickness of connecting lines. The structure of the unchanged part $\widetilde{\mathbf{X}}_i$ in optical image can be well conformed by the $\widetilde{\mathbf{Y}}_i$ in SAR image, while the structure of the changed part $\widetilde{\mathbf{X}}_i$ can not be conformed by the $\widetilde{\mathbf{Y}}_i$.

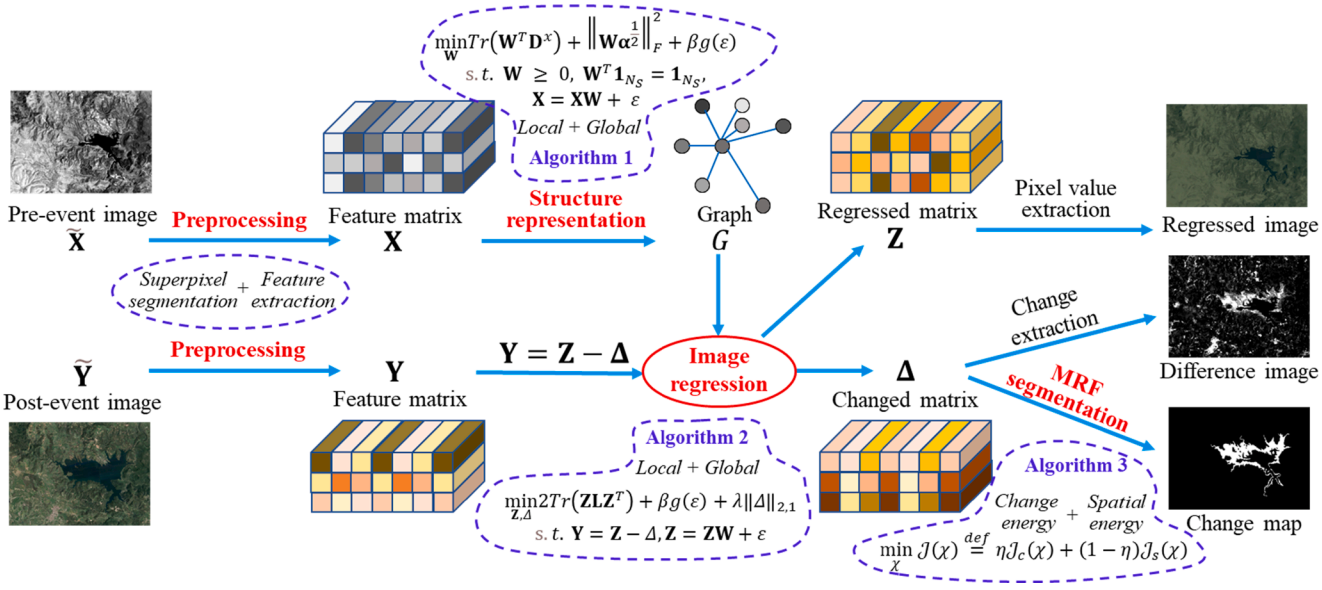


Fig. 3. Framework of the proposed heterogeneous change detection method.

can be extracted to represent the superpixel, such as the spatial, spectral (intensity), and textural information. In this paper, the mean, median, and variance values of each band are chosen as the spectral features for simplicity (other discriminatory features could also be used). Thus, we can obtain the feature matrix of each image as $\mathbf{X} \in \mathbb{R}^{3C_X \times N_S}$ and $\mathbf{Y} \in \mathbb{R}^{3C_Y \times N_S}$, and the columns of $\mathbf{X}_i$ and $\mathbf{Y}_i$ represent the feature vectors of superpixels $\widetilde{\mathbf{X}}_i$ and $\widetilde{\mathbf{Y}}_i$, respectively.

3.2. Structure representation by constructing graph

In order to complete the structure consistency-based image regression, we first construct a weighted directed graph $G = (V, E, \boldsymbol{w})$ to capture the structure information of image $\widetilde{\mathbf{X}}$, where the vertex set $V = \left\{\widetilde{\mathbf{X}}_1, \cdots, \widetilde{\mathbf{X}}_{N_S}\right\}$. If there is an edge $\left(\widetilde{\mathbf{X}}_i, \widetilde{\mathbf{X}}_j\right) \in E$ from $\widetilde{\mathbf{X}}_i$ to $\widetilde{\mathbf{X}}_j$, the entry $w_{i,j}$ represents the weight of this edge; otherwise, $w_{i,j} = 0$.

3.2.1. Model formulation

The local structure learning model (2) can capture the similarity relationships between each superpixel and its k-nearest neighbors, and the global structure learning model (3) can obtain the self-expression relationship between each superpixel and other superpixels that satisfy the constraint of $\mathbf{X}_i \approx \sum_{j=1}^{N} \mathbf{X}_j w_{j,i}$. To make use of complementary information provided by the local structure and global structure, we combine (2) and (3) into a unified objective function as

$$\begin{aligned} &\min_{\mathbf{W}} \underbrace{\sum_{i=1}^{N_S}\sum_{j=1}^{N_S} \left\|\mathbf{X}_i - \mathbf{X}_j\right\|_2^2 w_{i,j}}_{\text{local structure}} + \underbrace{\sum_{i=1}^{N_S} \alpha_i \|\mathbf{W}_i\|_2^2}_{\text{regularizer}} + \underbrace{\beta g(\boldsymbol{\varepsilon})}_{\text{global structure}} \\ &s.t.\ 0 \leqslant w_{i,j} \leqslant 1,\ \sum_{i=1}^{N} w_{i,j} = 1,\ \mathbf{X} = \mathbf{X}\mathbf{W} + \boldsymbol{\varepsilon}, \end{aligned} \tag{7}$$

where $\alpha_i, \beta > 0$ are the balancing parameters, $\boldsymbol{\varepsilon}$ represents the unknown corruption (self-expression error). $g(\boldsymbol{\varepsilon})$ represents the penalty term, which can be the squared Frobenius norm, ℓ_1-nrom, or $\ell_{2,1}$-nrom. Specifically, $\|\boldsymbol{\varepsilon}\|_F^2$ is more appropriate when the corruption obeys a Gaussian distribution; $\|\boldsymbol{\varepsilon}\|_1$ is usually adopted for random impulse noise; and $\|\boldsymbol{\varepsilon}\|_{2,1}$ is more suitable to characterize sample-specific corruption and outliers.

From (7), we can find that $\mathbf{W}$ can be regarded as a probability matrix,

that is, for the j-th superpixel $\widetilde{\mathbf{X}}_j$, all the superpixels can be connected to it as neighbors with probabilities $w_{i,j}, i = 1, \cdots, N_S$ with $0 \leqslant w_{i,j} \leqslant 1$, $\sum_{i=1}^{N_S} w_{i,j} = 1$. For the local structure penalty term, it prompts to assign a larger probability $w_{i,j}$ to $\widetilde{\mathbf{X}}_i$ when distance between $\mathbf{X}_i$ and $\mathbf{X}_j$ is smaller. For the global self-expression penalty term, it prompts to reconstruct $\mathbf{X}_j$ from the whole $\mathbf{X}$ with $\mathbf{W}_j$ as $\mathbf{X}_j \approx \mathbf{X}\mathbf{W}_j$. For the regularizer of $\alpha_i \|\mathbf{W}_i\|_2^2$, it is a smooth term to avoid trivial solution and make $\mathbf{W}$ sparse together with the condition $0 \leqslant w_{i,j} \leqslant 1, \sum_{i=1}^{N_S} w_{i,j} = 1$. Specially, if we set $\alpha_i = 0$, problem (7) have a trivial solution for $\mathbf{W}_i$ with $w_{i,i} = 1$ and $w_{j,i} = 0$, $j \neq i$, that is, $\widetilde{\mathbf{X}}_i$ is only connected to itself with probability 1. On the other hand, if we choose $\alpha_i \to \infty$, the optimal solution of $\mathbf{W}_i$ is that $\mathbf{W}_i = \mathbf{1}_{N_S}/N_S$, that is, all the superpixels connect $\widetilde{\mathbf{X}}_i$ with the same probability $1/N_S$. By combining the local structure and global structure penalty terms, the $\mathbf{W}$ in (7) can contain more information and the learned graph G is more representative than the local structure induced graph in Sun et al., 2021a, 2021b and the global structure induced graph in Sun et al. (2020).

3.2.2. Optimization

Define the distance matrix $\mathbf{D}^x \in \mathbb{R}^{N_S \times N_S}$ with the element being $d_{i,j}^x = \|\mathbf{X}_i - \mathbf{X}_j\|_2^2$, then we have

$$\mathbf{D}^x = diag\left(\mathbf{X}^T\mathbf{X}\right)\mathbf{1}_{N_S}\mathbf{1}_{N_S}^T + \mathbf{1}_{N_S}\mathbf{1}_{N_S}^T diag\left(\mathbf{X}^T\mathbf{X}\right) - 2\mathbf{X}^T\mathbf{X}, \tag{8}$$

where $diag\left(\mathbf{X}^T\mathbf{X}\right)$ is a diagonal matrix with the diagonal elements of $\mathbf{X}^T\mathbf{X}$. Define the $\boldsymbol{\alpha}^{\frac{1}{2}} \in \mathbf{R}^{N_S \times N_S}$ as a diagonal matrix with diagonal elements being $\sqrt{\alpha_i}$. Then, problem (7) can be rewritten as

$$\begin{aligned} &\min_{\mathbf{W}} Tr\left(\mathbf{W}^T\mathbf{D}^x\right) + \left\|\mathbf{W}\boldsymbol{\alpha}^{\frac{1}{2}}\right\|_F^2 + \beta g(\boldsymbol{\varepsilon}) \\ &\quad s.t.\ \ \mathbf{W} \geqslant 0,\ \ \mathbf{W}^T\mathbf{1}_{N_S} = \mathbf{1}_{N_S},\ \ \mathbf{X} = \mathbf{X}\mathbf{W} + \boldsymbol{\varepsilon}. \end{aligned} \tag{9}$$

Problem (9) can be efficiently solved by using the alternating direction method of multipliers (ADMM) in the Appendix A, and the detailed derivation is provided in the supplementary document. The procedure of solving the problem (9) is summarized in Algorithm 1 of Table 2. The algorithm terminates when the maximal number of iterations N_{iter} is reached or the relative difference between two iteration results $\xi^{(t+1)} < \xi^0$, where $\xi^{(t+1)} = \frac{\|\mathbf{W}^{(t+1)} - \mathbf{W}^{(t)}\|_F}{\|\mathbf{W}^{(t+1)}\|_F}$.

3.2.3. k selection

From the process of solving $\mathbf{W}$ in Appendix A, we can find that the constructed graph $G = (V, E, w)$ is a KNN type graph with the probabilistic weight $\mathbf{W}$ of (32c) and the regularization parameter α_i can be replaced by number of neighbors k_i as proven in the supplementary material, which plays an important role in the graph G. Obviously, a very small k_i is not robust enough for the graph, whereas a very large k_i tends to over connected the graph and leads to confusion. Therefore, we need to choose a suitable k_i for each vertex instead of a common k for all vertexes. For each superpixel, we want "*it to be connected and reconstructed by as many truly similar superpixels as possible*". Here, we propose a strategy to adaptively select k_i for each vertex by using the in-degree of the vertex.

Step 1. Set $k_{\max} = \sqrt{N_S}$ and $k_{\min} = \sqrt{N_S}/10$, and find the $k_{\max}$ nearest-neighbors of each vertex.

Step 2. Calculate the in-degree $di\left(\widetilde{\mathbf{X}}_i\right)$ for each vertex $\widetilde{\mathbf{X}}_i$, that is, compute the number of times $\widetilde{\mathbf{X}}_i$ occurs among the $k_{\max}$ nearest-neighbors of all vertexes.

Step 3. Choose $k_i = \min\left\{k_{\max}, \max\left\{di\left(\widetilde{\mathbf{X}}_i\right), k_{\min}\right\}\right\}$ for each vertex.

With this strategy of k selection, we can select a smaller k_i for superpixel that belongs to low density (has few truly similar superpixels), and select a larger k_i for superpixel that belongs to high density (has many truly similar superpixels).

Table 2
Implementation steps of Algorithm 1.

Algorithm 1. Structured graph learning
Input: The feature matrix $\mathbf{X}$, parameter $\beta > 0$.
Initialize: Set $\mathbf{S}$, $\mathbf{R}_1$, and $\mathbf{R}_2 = 0$, and adaptively select k_i.
Repeat:
1: Update $\boldsymbol{\varepsilon}^{(t+1)}$ through Eq. (32a) according to different $g(\cdot)$.
2: Update $\mathbf{S}^{(t+1)}$ through Eq. (32b).
3: Update $\mathbf{W}^{(t+1)}$ through Eq. (32c).
4: Update the Lagrangian multipliers through Eq. (32d) and (32e).
Until stopping criterion is met.
Output: The learned probability matrix $\mathbf{W}$.

3.2.4. Hypergraph construction

With the learned probability matrix $\mathbf{W}$, we can construct the hypergraph $G^h = \{V^h, E^h, w^h\}$. With $V^h = \left\{\widetilde{\mathbf{X}}_1, \cdots, \widetilde{\mathbf{X}}_{N_S}\right\}$, each superpixel $\widetilde{\mathbf{X}}_i$ corresponds to a hyperedge e_i, i.e., we treat $\widetilde{\mathbf{X}}_i$ as a center, and connect $\widetilde{\mathbf{X}}_i$ and its neighbors in $\mathbf{W}_i$ to generate a hyperedge $e_i = \left\{\widetilde{\mathbf{X}}_i\right\} \cup \left\{\widetilde{\mathbf{X}}_j \middle| w_{j,i} \neq 0, j = 1, \cdots, N_S\right\}$. As proved in the supplementary material, each vertex in graph G has a loop (each vertex is connected to itself), i.e., $w_{i,i} > 0$ for $i = 1, \cdots, N_S$. Then, we have the hyperedge set $E^h = \{e_i | i = 1, \cdots, N_S\}$ with e_i defined as

$$e_i = \left\{\widetilde{\mathbf{X}}_j \middle| w_{j,i} \neq 0; j = 1, \cdots, N_S\right\}. \tag{10}$$

For the traditional hypergraph graph construction method, the incident matrix $\mathbf{H} \in \mathbb{R}^{N_S \times N_S}$ on the hypergraph can be represented as $h(v_i, e_j) = \begin{cases} 1, & \text{if } w_{i,j} \neq 0 \\ 0, & \text{otherwise} \end{cases}$.

Assume that $\widetilde{\mathbf{X}}_j$ and $\widetilde{\mathbf{X}}_l$ are both contained in e_i, where the center point $\widetilde{\mathbf{X}}_i$ of e_i have different distances with $\widetilde{\mathbf{X}}_j$ and $\widetilde{\mathbf{X}}_l$, respectively. Thus $\widetilde{\mathbf{X}}_j$ and $\widetilde{\mathbf{X}}_l$ should assign different weights in e_i, respectively. However, the traditional incident matrix $\mathbf{H}$ can not meet this demand since it is binary. Actually, in the probability matrix $\mathbf{W}$, each element $w_{j,i} > 0$ gives a weight to the superpixel $\widetilde{\mathbf{X}}_j$ in e_i. Furthermore, we set the incident matrix of G^h as $\widetilde{\mathbf{H}} = \mathbf{W}$, and choose the mean of the heat kernel weights of the intra-class superpixels as the hyperedge weight $w^h(e)$

$$w^h(e_i) = \frac{1}{k_i(k_i - 1)} \sum_{\widetilde{\mathbf{X}}_j, \widetilde{\mathbf{X}}_l \in e_i} \exp\left(-\frac{\|\mathbf{X}_j - \mathbf{X}_l\|_2^2}{\sigma}\right), \tag{11}$$

where $\sigma = \frac{1}{k_i(k_i-1)} \sum_{\widetilde{\mathbf{X}}_j, \widetilde{\mathbf{X}}_l \in e_i} \|\mathbf{X}_j - \mathbf{X}_l\|_2^2$ denotes the mean distance among all the neighbor superpixels in the hyperedge. Using this weight $w^h(e)$, the compact hyperedge with small average intra-class distance is assigned a higher hyperedge weight. Then we can obtain the diagonal hyperedge weight matrix $\mathbf{W}^h$ with the diagonal element $w_{i,i}^h = w^h(e_i)$. Subsequently, we rewrite the hypergraph Laplacian matrix as

$$\mathbf{L}^h = \mathbf{D}_v - \widetilde{\mathbf{H}}\mathbf{W}^h\mathbf{D}_e^{-1}\widetilde{\mathbf{H}}^T. \tag{12}$$

3.3. Structure consistency based image regression

Once the matrix $\mathbf{W}$ that represents the structure information of pre-event image is obtained, the image regression can be completed by using structure consistency. For example, we can project $\mathbf{W}$ into the domain of post-event $\mathbf{Y}$ to obtain the regressed feature matrix $\mathbf{Z} = \mathbf{Y}\mathbf{W}$, which can be regarded as using the structure of $\widetilde{\mathbf{X}}_i$ to translate $\widetilde{\mathbf{X}}_i$ in the $\mathscr{Y}$ domain as $\mathbf{Z}_i = \sum_{j=1}^{N_S} \mathbf{Y}_j w_{j,i}$. Then, the regression image $\widetilde{\mathbf{Z}}$ can be calculated by

extracting the mean features in $\mathbf{Z}$ as

$$\widetilde{z}(m,n,c) = z_{c,i},\ \ (m,n) \in \Lambda_i, \tag{13}$$

where $\widetilde{z}(m,n,c)$ is the pixel value of $\widetilde{\mathbf{Z}}$ and $z_{c,i}$ is the mean feature of each band in $\mathbf{Z}$.

However, directly translating the i-th superpixel without considering the stability of neighbors (changed or unchanged) is not appropriate. That is, the changed neighbors will affect the regression performance. For example, for the unchanged target i-th superpixel $\widetilde{\mathbf{X}}_i$, although it can be represented by $\mathbf{X}_i = \sum_{j=1}^{N_S} \mathbf{X}_j w_{j,i}$, directly using $\mathbf{Z}_i = \sum_{j=1}^{N_S} \mathbf{Y}_j w_{j,i}$ to translate $\mathbf{X}_i$ will cause errors when some neighbors $\widetilde{\mathbf{X}}_j$ with $w_{j,i} > 0$ are changed in the event.

Therefore, to alleviate the negative effects of changed superpixels, we decompose the post-event image $\widetilde{\mathbf{Y}}$ into the regression image $\widetilde{\mathbf{Z}}$ of the pre-event image and the changed image $\widetilde{\mathbf{\Delta}}$ as $\widetilde{\mathbf{Y}} = \widetilde{\mathbf{Z}} - \widetilde{\mathbf{\Delta}}$. Then we have $\mathbf{Y} = \mathbf{Z} - \mathbf{\Delta}$, where $\mathbf{\Delta} \in \mathbb{R}^{3C_Y \times N_S}$ is the changed feature matrix ($\mathbf{\Delta}$ is not the feature matrix of $\widetilde{\mathbf{\Delta}}$ since the feature extraction is not a linear operator).

3.3.1. Structure consistency based regularization

As illustrated in Fig. 2, the structure consistency between $\widetilde{\mathbf{X}}$ and its regression image $\widetilde{\mathbf{Z}}$ requires that they should share the same similarity relationships, which contains two types of constraints that correspond to the graph learning process: one is the global self-expression constraint as $\mathbf{Z} = \mathbf{ZW}$ and the other is the local similarity constraint, that is if superpixels in the pre-event images ($\widetilde{\mathbf{X}}_i$ and $\widetilde{\mathbf{X}}_j$) are very similar, then the superpixels in the regression image ($\widetilde{\mathbf{Z}}_i$ and $\widetilde{\mathbf{Z}}_j$) corresponding to this superpixel pair should also be similar. Since the structure information (similarity relationships) of $\widetilde{\mathbf{X}}$ is represented by the graph G and G^h, we have two following constraints based on two local constraint strategies.

GLR: we desire the regressed superpixels corresponding to the instances connected in the edge of G be similar to each other. Specifically, the GLR is given by

$$\min_{\mathbf{Z}} \sum_{i=1}^{N_S} \sum_{j=1}^{N_S} \|\mathbf{Z}_i - \mathbf{Z}_j\|_2^2 w_{i,j}. \tag{14}$$

Denote the degree matrix $\mathbf{D}^g \in \mathbb{R}^{N_S \times N_S}$ of graph G as a diagonal matrix with the i-th entry $D^g_{i,i}$ corresponds to the summation of all the similarities related to $\widetilde{\mathbf{X}}_i$, i.e., $D^g_{i,i} = \sum_{j=1}^{N_s} (w_{i,j} + w_{j,i})/2$, and the graph Laplacian matrix is defined as

$$\mathbf{L}^g = \mathbf{D}^g - \frac{\mathbf{W} + \mathbf{W}^T}{2}. \tag{15}$$

Then, we have

$$\sum_{i=1}^{N_S} \sum_{j=1}^{N_S} \|\mathbf{Z}_i - \mathbf{Z}_j\|_2^2 w_{i,j} = 2Tr(\mathbf{ZL}^g\mathbf{Z}^T). \tag{16}$$

HGLR: we desire the regressed superpixels corresponding to the instances within the same hyperedge of G^h be similar to each other. Specifically, the HGLR is given by

$$\sum_{e \in E^h} \sum_{\{\widetilde{\mathbf{X}}_i, \widetilde{\mathbf{X}}_j\} \in e} \frac{w^h(e)\widetilde{h}(\widetilde{\mathbf{X}}_i, e)\widetilde{h}(\widetilde{\mathbf{X}}_j, e)}{\delta(e)} \|\mathbf{Z}_i - \mathbf{Z}_j\|_2^2 = 2Tr(\mathbf{ZL}^h\mathbf{Z}^T). \tag{17}$$

The detailed derivation of (17) is in the supplementary document. From the HGLR constraint of (17), we can find that the similarity among the superpixels within the same hyperedge of G^h is kept by the regressed $\widetilde{\mathbf{Z}}$.

Combining the global self-expression constraint and the local similarity constraint, we can obtain the structure consistency based regularization as

$$\min_{\mathbf{Z}} 2Tr(\mathbf{ZLZ}^T) + \beta g(\mathbf{Z} - \mathbf{ZW}), \tag{18}$$

where $\beta > 0$ is a balancing parameter, and the Laplacian matrix $\mathbf{L}$ can be the graph Laplacian matrix $\mathbf{L}^g$ in (15) or the hypergraph Laplacian matrix $\mathbf{L}^h$ in (12) according to different local constraint strategies. The penalty function $g(\cdot)$ is similar to the function in (7), which can be the squared Frobenius norm, ℓ_1-nrom, or $\ell_{2,1}$-nrom according to the statistical properties of the self-expression errors.

3.3.2. Objective function

Based on the fact that most of the objects are unchanged and only a small part of objects are changed in the event, we can use the prior sparsity together with the structure consistency based regularization to model the regression process as

$$\min_{\mathbf{Z},\mathbf{\Delta}} 2Tr(\mathbf{ZLZ}^T) + \beta g(\mathbf{Z} - \mathbf{ZW}) + \lambda\|\mathbf{\Delta}\|_{2,1} s.t.\ \ \mathbf{Y} = \mathbf{Z} - \mathbf{\Delta}, \tag{19}$$

where $\lambda > 0$ is a penalty parameter. The $\ell_{2,1}$-norm regularization term $\|\mathbf{\Delta}\|_{2,1}$ is a convex relaxation of the $\ell_{2,0}$-norm of $\|\mathbf{\Delta}\|_{2,0}$, which is used to promote column sparseness.

3.3.3. Optimization

By using the ADMM and introducing an auxiliary variable $\boldsymbol{\varepsilon} \in \mathbb{R}^{3C_Y \times N_S}$, the model (19) can be rewritten as

$$\begin{aligned} &\min_{\mathbf{Z},\mathbf{\Delta}} 2Tr(\mathbf{ZLZ}^T) + \beta g(\boldsymbol{\varepsilon}) + \lambda\|\mathbf{\Delta}\|_{2,1} \\ &s.t.\ \ \mathbf{Y} = \mathbf{Z} - \mathbf{\Delta},\ \ \mathbf{Z} = \mathbf{ZW} + \boldsymbol{\varepsilon}. \end{aligned} \tag{20}$$

Problem (20) can be efficiently solved by using the ADMM in the Appendix B. The framework of solving minimization problem (20) is summarized in Table 3 (Algorithm 2). The stopping criterion of Algorithm 2 is that the maximum number of iterations N_{iter} is reached or relative difference between two iteration results $\xi^{(t+1)} < \xi^0$, where $\xi^{(t+1)} = \frac{\|\mathbf{\Delta}^{(t+1)} - \mathbf{\Delta}^{(t)}\|_F}{\|\mathbf{\Delta}^{(t+1)}\|_F}$.

Once the regression feature matrix $\mathbf{Z}$ is computed from Algorithm 2, the regression image $\widetilde{\mathbf{Z}}$ can be obtained by extracting the mean features in $\mathbf{Z}$ as in (13). With the changed feature matrix $\mathbf{\Delta}$ output by Algorithm 2, we can compute the DI as

$$DI(m,n) = \|\mathbf{\Delta}_i\|_2^2,\ \ (m,n) \in \Lambda_i. \tag{21}$$

3.4. Binary CM calculation with MRF segmentation

Once the sparse DI is obtained, the MCD problem can be regarded as an image binary segmentation problem, which can be solved by thresholding method or clustering method. Here, we treat the binary CM calculation as a superpixel-labeling problem, which assigns a label χ_i for the i-th superpixel with $\chi_i = 0$ representing that the region of Λ_i is unchanged and $\chi_i = 1$ representing that Λ_i is changed in the event. Define $\boldsymbol{\chi} = \{\chi_i | i = 1, \cdots, N_S\}$ as the label set of superpixels, the superpixel-

Table 3
Implementation steps of Algorithm 2.

Algorithm 2. Structure consistency based image regression.
Input: The matrices of $\mathbf{Y}$, $\mathbf{W}$ and $\mathbf{L}$, parameters $\beta, \lambda > 0$.
Initialize: Set $\mathbf{Z}$, $\mathbf{R}_1$, and $\mathbf{R}_2 = \mathbf{0}$.
Repeat:
1: Update $\boldsymbol{\varepsilon}^{(t+1)}$ through Eq. (35a) according to different $g(\cdot)$.
2: Update $\mathbf{\Delta}^{(t+1)}$ through Eq. (35b).
3: Update $\mathbf{Z}^{(t+1)}$ through Eq. (35c).
4: Update the Lagrangian multipliers through Eqs. (35d) and (35e).
Until stopping criterion is met.
Output: The regressed feature matrix $\mathbf{Z}$ and changed feature matrix $\mathbf{\Delta}$.

labeling is equivalent to minimizing an energy function $\mathscr{J}(\chi)$, which is the log likelihood of the posterior distribution of MRF (Geman and Geman, 1984; Szeliski et al., 2008)

$$\min_{\chi} \mathscr{J}(\chi) \overset{def}{=} \eta \mathscr{J}_c(\chi) + (1-\eta)\mathscr{J}_s(\chi), \tag{22}$$

where $\mathscr{J}_c(\chi)$ denotes the change energy term, $\mathscr{J}_s(\chi)$ denotes the spatial energy term, and $\eta \in (0,1)$ is a balancing parameter.

3.4.1. Change energy term

$\mathscr{J}_c(\chi)$ is often model as two Gaussian mixture models (GMMs) (Rother et al., 2004), one for the unchanged class and one for the changed class, are taken to be a full-covariance Gaussian mixture with $\mathscr{K}$ components (we set $\mathscr{K} = \min\{3C_{\mathbf{Y}}, 10^{-3}N_S\}$ in this paper for simplicity). In order to handle GMM concisely, we introduce an additional vector $\boldsymbol{\kappa} = \{\kappa_i | i = 1, \cdots, N_S\}$ with $\kappa_i \in \{1, \cdots, \mathscr{K}\}$, which assigns a unique GMM component to each superpixel. The component is either from the unchanged or the changed class according to $\chi_i = 0$ or 1. With the GMM, the change energy term $\mathscr{J}_c(\chi)$ is defined as

$$\mathscr{J}_c(\chi) = \min_{\boldsymbol{\kappa}} \sum_{i=1}^{N_S} \varphi_c(\chi_i, \kappa_i, \theta, \boldsymbol{\Delta}_i), \tag{23}$$

where $\varphi_c(\chi_i, \kappa_i, \theta, \boldsymbol{\Delta}_i) = -\log p(\boldsymbol{\Delta}_i | \chi_i, \kappa_i, \theta) - \log \pi_{(\chi_i, \kappa_i)}$ and $p(\cdot)$ is a Gaussian probability distribution, and $\pi_{(\cdot)}$ is the mixture weighting coefficient, then we have

$$\begin{aligned} \varphi_c(\chi_i, \kappa_i, \theta, \boldsymbol{\Delta}_i) = & -\log \pi_{(\chi_i, \kappa_i)} + \frac{1}{2}\log\det\Sigma_{(\chi_i, \kappa_i)} \\ & + \frac{1}{2}\left(\boldsymbol{\Delta}_i - \mu_{(\chi_i, \kappa_i)}\right)^T \Sigma^{-1}_{(\chi_i, \kappa_i)} \left(\boldsymbol{\Delta}_i - \mu_{(\chi_i, \kappa_i)}\right), \end{aligned} \tag{24}$$

and the parameter θ of the model is

$$\theta = \left\{ \pi_{(\chi,\kappa)}, \mu_{(\chi,\kappa)}, \Sigma_{(\chi,\kappa)} \middle| \chi = 0, 1; \kappa = 1, \cdots, \mathscr{K} \right\}. \tag{25}$$

That is the weights π, means μ and covariances Σ of the $2\mathscr{K}$ Gaussian components for the distributions of unchanged and changed classes. For a given GMM component κ, for example, in the unchanged class, the subset of superpixels $\mathscr{S}(0,\kappa) = \{\boldsymbol{\Delta}_i | \chi_i = 0, \kappa_i = \kappa\}$ is defined. Then the parameter in θ can be estimates as: $\pi_{(0,\kappa)} = |\mathscr{S}(0,\kappa)| / \sum_{\kappa=1}^{\mathscr{K}} |\mathscr{S}(0,\kappa)|$, and the mean $\mu_{(0,\kappa)}$ and covariance $\Sigma_{(0,\kappa)}$ are estimated in standard fashion as the sample mean and covariance of superpixels in $\mathscr{S}(0,\kappa)$.

3.4.2. Spatial energy term

An R-adjacency neighbor system is constructed for $\mathscr{J}_s(\chi)$, that is, if two superpixels (located in Λ_i and Λ_j) intersect or the distance between their center points is less than R, these two superpixels are marked to be the R-adjacency neighbors of each other denoted as $i \in \mathscr{N}_j^R$ (or $j \in \mathscr{N}_i^R$). As the size of each superpixel generated by SLIC is around MN/N_S, we set $R = 2\sqrt{MN/N_S}$ for simplicity.

Instead of using the commonly used spatial smoothness term built on the DI, we construct a novel spatial energy term $\mathscr{J}_s(\chi)$, which not only takes into account the spatial information of DI but also the similarity information of the original pre- and post-event images. It is defined as

$$\mathscr{J}_s(\chi) = \sum_{i=1}^{N_S} \sum_{j \in \mathscr{N}_i^R} \frac{\varphi_s(\chi_i, \chi_j)\delta(\chi_i - \chi_j)}{d(\Lambda_i, \Lambda_j)}, \tag{26}$$

where $\delta(\cdot)$ is the function defined as $\delta(x) = 1$ if $x \neq 0$ and $\delta(x) = 0$ if $x = 0$, and $d(\Lambda_i, \Lambda_j)$ is the Euclidean spatial distance between two superpixels. The function $\varphi_s(\chi_i, \chi_j)$ is defined as

$$\varphi_s(\chi_i, \chi_j) = \begin{cases} \exp\left(-\frac{d^x_{i,j}}{2\sigma_1^2}\right)\exp\left(-\frac{d^y_{i,j}}{2\sigma_2^2}\right), & \text{if } d^x_{i,j} \leqslant \sigma_1^2, d^y_{i,j} \leqslant \sigma_2^2 \\ \exp\left(\frac{d^x_{i,j}}{2\sigma_1^2} - 1\right)\exp\left(-\frac{d^y_{i,j}}{2\sigma_2^2}\right), & \text{if } d^x_{i,j} \leqslant \sigma_1^2, d^y_{i,j} > \sigma_2^2 \\ \exp\left(-\frac{d^x_{i,j}}{2\sigma_1^2}\right)\exp\left(\frac{d^y_{i,j}}{2\sigma_2^2} - 1\right), & \text{if } d^x_{i,j} > \sigma_1^2, d^y_{i,j} \leqslant \sigma_2^2 \\ \exp(-1), & \text{if } d^x_{i,j} > \sigma_1^2, d^y_{i,j} > \sigma_2^2 \end{cases}, \tag{27}$$

where $d^x_{i,j} = \|\mathbf{X}_i - \mathbf{X}_j\|_2^2, d^y_{i,j} = \|\mathbf{Y}_i - \mathbf{Y}_j\|_2^2$ represent the feature distances between superpixels in the pre- and post-event images, respectively. $\sigma_1^2 = \frac{\sum_{i=1}^{N_S}\sum_{j \in \mathscr{N}_i^R} d^x_{i,j}}{\sum_{i=1}^{N_S} |\mathscr{N}_i^R|}$, $\sigma_2^2 = \frac{\sum_{i=1}^{N_S}\sum_{j \in \mathscr{N}_i^R} d^y_{i,j}}{\sum_{i=1}^{N_S} |\mathscr{N}_i^R|}$ are the normalization parameters that represent the average feature distances.

From this spatial energy term in (26) and (27), we can find that it contains four cases: **1)** when $\mathbf{X}_i$ and $\mathbf{X}_j$, $\mathbf{Y}_i$ and $\mathbf{Y}_j$ are similar (i.e., $d^x_{i,j}$ and $d^y_{i,j}$ are very small), then the probability that labels of the i-th superpixel χ_i and j-th superpixel χ_j are the same should be high. And as $d^x_{i,j}$ and $d^x_{i,j}$ decrease, the probability of $\chi_i = \chi_j$ increases. Then $\varphi_s(\chi_i, \chi_j)$ gives a greater penalty for the discontinuity of $\chi_i \neq \chi_j$ under this case. **2)** when $\mathbf{X}_i$ and $\mathbf{X}_j$ are similar and $\mathbf{Y}_i$ and $\mathbf{Y}_j$ are not similar (that is $d^x_{i,j}$ is small and $d^y_{i,j}$ is large), then the probability that labels of χ_i and χ_j are the same should be low. And as $d^x_{i,j}$ decreases and $d^y_{i,j}$ increases, the probability of $\chi_i = \chi_j$ decreases. Then $\varphi_s(\chi_i, \chi_j)$ gives a smaller penalty for the discontinuity of $\chi_i \neq \chi_j$ under this case. **3)** when $\mathbf{X}_i$ and $\mathbf{X}_j$ are not similar and $\mathbf{Y}_i$ and $\mathbf{Y}_j$ are similar (that is $d^x_{i,j}$ is large and $d^y_{i,j}$ is small), similar to the second case, $\varphi_s(\chi_i, \chi_j)$ gives a small penalty for the discontinuity of $\chi_i \neq \chi_j$ under this case. **4)** when $\mathbf{X}_i$ and $\mathbf{X}_j$, $\mathbf{Y}_i$ and $\mathbf{Y}_j$ are not similar (that is $d^x_{i,j}$ and $d^y_{i,j}$ are large), it means that the i-th superpixel and j-th superpixel are not closely related to each other, that is, the relationship between their labels is also ambiguous. Then $\varphi_s(\chi_i, \chi_j)$ gives a median discontinuity penalty for this case.

3.4.3. Graph cut

With defined change energy term $\mathscr{J}_c(\chi)$ in (23) and spatial energy term $\mathscr{J}_c(\chi)$ in (27), the energy minimization problem of (22) can be solved by using the min-cut/max-flow algorithm (Boykov and Kolmogorov, 2004).

The iterative framework for the GMM based MRF segmentation problem is summarized in Table 4 (Algorithm 3). The initial χ is obtained by using the Otsu thresholding method (Otsu, 1979) on the DI calculated by (21). Step 1 is completed directly by a simple enumeration of the κ_i values (from 1 to $\mathscr{K}$) for each superpixel. Step 2 is a process of estimating a set of Gaussian parameters θ, as previously described. Step 3 is completed by using the graph cut method proposed in Boykov and Kolmogorov (2004).

Once the final χ is assigned, we can obtain the binary CM as

$$CM(m,n) = \chi_i, \ (m,n) \in \Lambda_i. \tag{28}$$

The overall framework of the proposed graph based image regression and MRF segmentation method for MCD problem is summarized in Table 5 (called (H) GIR-MRF for short, with H standing for the hypergraph based local constraint), which mainly contains four processes: preprocessing, structured graph learning (Algorithm 1), image regression (Algorithm 2), and MRF segmentation (Algorithm 3).

Table 4
Implementation steps of Algorithm 3.

Algorithm 3. GMM based MRF segmentation.
Input: The matrices of $\mathbf{\Delta}$, $\mathbf{X}$ and $\mathbf{Y}$, parameter $\eta > 0$.
Initialize: Calculate χ by Otsu thresholding method.
Initialize $\boldsymbol{\kappa}$ by using kmeans clustering on χ.
Estimate the Gaussian parameters θ.
Repeat:
1: Assign GMM component to each superpixel:
$\kappa_i^{(t+1)} = \underset{\kappa_i}{\operatorname{argmin}} \varphi_c\left(\chi_i^{(t)}, \kappa_i, \theta^{(t)}, \mathbf{\Delta}_i\right)$.
2: Estimate the Gaussian parameters $\theta^{(t+1)}$ with $\boldsymbol{\kappa}^{(t+1)}$ and $\chi^{(t)}$.
3: Use graph cut to solve problem of Eq. (22) with $\boldsymbol{\kappa}^{(t+1)}$ and $\theta^{(t+1)}$:
$\chi^{(t+1)} = \underset{\chi}{\operatorname{argmin}} J(\chi)$.
Until stopping criterion is met.
Output: The label set χ.

Table 5
Framework of (H) GIR-MRF.

(H) GIR-MRF.
Input: Images of $\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{Y}}$, parameters of N_S, β, λ, and η.
Preprocessing:
Implement the (modified) SLIC on $\widetilde{\mathbf{X}}$ to obtain Λ.
Extract the features to obtain $\mathbf{X}$ and $\mathbf{Y}$.
Structured graph leaning:
Compute the probability matrix $\mathbf{W}$ by using Algorithm 1.
Construct the hypergraph G^h if necessary.
Image regression:
Construct the graph/hypergraph Laplacian matrix $\mathbf{L}$.
Compute the changed feature matrix $\mathbf{\Delta}$ by using Algorithm 2.
MRF segmentation:
Compute the label set χ by using Algorithm 3.
Compute the binary change map with Eq. (28).

4. Experiments and discussions

In this section, experiments are conducted to evaluate the performance of the proposed (H) GIR-MRF. The illustration of data sets is firstly presented. Then the regression performance and CD performance are demonstrated. Following that, some detailed discussions about the parameters and computational complexity are made.

4.1. Data sets, evaluation metrics and parameters setting

Six heterogeneous data sets are presented to evaluate the propose method as listed in Table 6. These data sets contain different types of multimodalities: multisensor optical images (same sensor type but with different sensors, e.g., #1 to #4) and multisource images (different sensor types, e.g., #5 and #6), provide different resolutions (varying from 0.52 to 30 m), cover different image sizes (varying from 300 to 2000 pixels in width or length), and reflecting different types of events (such as flooding, fire, and construction), which can evaluate the robustness of the proposed method in different CD conditions.

To evaluate the performance of DI generated by the proposed method, the empirical receiver operating characteristics (ROC) curve is plotted, and the corresponding area under the curve (AUC) is used as the quantitative criterion. To assess the final CM generated by (H) GIR-MRF, three quantitative evaluation indices, overall accuracy (OA), Kappa coefficient (Kc) and F1 score (F1) are adopted as metrics.

For all the experiments of (H) GIR-MRF, we choose the penalty function $g(\boldsymbol{\varepsilon}) = \|\boldsymbol{\varepsilon}\|_F^2$ for Algorithm 1 and Algorithm 2; set $N_S = 10^4$ for the superpixel segmentation; fix $\beta = 1, \mu_1 = \mu_2 = 0.4, N_{iter} = 10, \xi^0 = 0.01$ for Algorithm 1 and Algorithm 2; set $\lambda = 0.01$ for Algorithm 2; and set $\eta = 0.025$ (except $\eta = 0.05$ for Dataset #1), $N_{iter} = 5$ for Algorithm 3. At the same time, since the initialization of GMM in the Algorithm 3 involves the use of kmeans clustering, which is randomized, we repeated Algorithm 3 fifty times to obtain the average segmentation performance.

4.2. Image regression performance of GIR-MRF

In the first experiment, we test GIR-MRF on the Datasets #1 and #2 to demonstrate the effectiveness of structured graph based image regression. Both Datasets #1 and #2 contain one pre-event image and two post-event images, as shown in Fig. 4(a)-(c). In Dataset #1, we denote the two NIR band images acquired in September 1995 and July 1996 as $\widetilde{\mathbf{X}}_{\mathrm{NIR}}^{t1}$ and $\widetilde{\mathbf{X}}_{\mathrm{NIR}}^{t2}$, respectively, and the optical image acquired in July 1996 as $\widetilde{\mathbf{Y}}_{\mathrm{opt}}^{t2}$. Similarly, in Dataset #2, we denote the two multispectral images acquired by Landsat-5 in August 2011 and September 2011 as $\widetilde{\mathbf{X}}_{\mathrm{L5}}^{t1}$ and $\widetilde{\mathbf{X}}_{\mathrm{L5}}^{t2}$, respectively, and the multispectral image acquired by the Advanced Land Image (ALI) from the Earth Observing (EO-1) mission in September 2011 as $\widetilde{\mathbf{Y}}_{\mathrm{ALI}}^{t2}$.

We translate the $\widetilde{\mathbf{X}}_{\mathrm{NIR}}^{t1}, \widetilde{\mathbf{X}}_{\mathrm{NIR}}^{t2}$ and $\widetilde{\mathbf{X}}_{\mathrm{L5}}^{t1}, \widetilde{\mathbf{X}}_{\mathrm{L5}}^{t2}$ to the domains of $\widetilde{\mathbf{Y}}_{\mathrm{opt}}^{t2}$ and $\widetilde{\mathbf{Y}}_{\mathrm{ALI}}^{t2}$ to obtain the regression images of $\widetilde{\mathbf{Z}}_{\mathrm{NIR}}^{t1}, \widetilde{\mathbf{Z}}_{\mathrm{NIR}}^{t2}$ and $\widetilde{\mathbf{Z}}_{\mathrm{L5}}^{t1}, \widetilde{\mathbf{Z}}_{\mathrm{L5}}^{t2}$, respectively, as shown in Figs. 4(d)-(e). By comparing Figs. 4(c) and 4(e), we can find that the structures of $\widetilde{\mathbf{Z}}_{\mathrm{NIR}}^{t2}$ and $\widetilde{\mathbf{Y}}_{\mathrm{opt}}^{t2}$, $\widetilde{\mathbf{Z}}_{\mathrm{L5}}^{t2}$ and $\widetilde{\mathbf{Y}}_{\mathrm{ALI}}^{t2}$ are consistent, i.e., the learned graphs G can well represent the structures of $\widetilde{\mathbf{X}}_{\mathrm{NIR}}^{t2}$ and $\widetilde{\mathbf{X}}_{\mathrm{L5}}^{t2}$, and be well conformed by the $\widetilde{\mathbf{Y}}_{\mathrm{opt}}^{t2}$ and $\widetilde{\mathbf{Y}}_{\mathrm{ALI}}^{t2}$, showing that the regression images of $\widetilde{\mathbf{Z}}_{\mathrm{NIR}}^{t2}$ and $\widetilde{\mathbf{Z}}_{\mathrm{L5}}^{t2}$ are very similar to the target domain images of $\widetilde{\mathbf{Y}}_{\mathrm{opt}}^{t2}$ and $\widetilde{\mathbf{Y}}_{\mathrm{ALI}}^{t2}$, respectively. On the other hand, by comparing Figs. 4(c) and 4(d), we can see that the structure consistency between images ($\widetilde{\mathbf{X}}_{\mathrm{NIR}}^{t1}$ and $\widetilde{\mathbf{Y}}_{\mathrm{opt}}^{t2}$, $\widetilde{\mathbf{X}}_{\mathrm{L5}}^{t1}$ and $\widetilde{\mathbf{Y}}_{\mathrm{ALI}}^{t2}$) is no longer maintained in the changed areas, showing that the regression images of $\widetilde{\mathbf{Z}}_{\mathrm{NIR}}^{t1}$ and $\widetilde{\mathbf{Z}}_{\mathrm{L5}}^{t1}$ are very different from the target domain images of $\widetilde{\mathbf{Y}}_{\mathrm{opt}}^{t2}$ and $\widetilde{\mathbf{Y}}_{\mathrm{ALI}}^{t2}$ in the region of changes, respectively. Fig. 4(f) shows the DIs generated by GIR-MRF with the pre-event $\widetilde{\mathbf{X}}_{\mathrm{NIR}}^{t1}, \widetilde{\mathbf{X}}_{\mathrm{L5}}^{t1}$ and post-event $\widetilde{\mathbf{Y}}_{\mathrm{opt}}^{t2}, \widetilde{\mathbf{Y}}_{\mathrm{ALI}}^{t2}$, respectively. It can be found that the DI can well measure the change level, showing the ability to detect changed areas.

4.3. CM performance of (H) GIR-MRF

To verify the effectiveness of the proposed method, we first compare

Table 6
Description of the six heterogeneous data sets.

Dataset	Sensor (or modality)	Size (pixels)	Date	Location	Event (& Spatial resolution)
#1	Landsat-5/Google Earth	$300 \times 412 \times 1(3)$	Sept. 1995 - July 1996	Sardinia, Italy	Lake expansion (30 m.)
#2	Landsat-5/EO-1 ALI	$1534 \times 808 \times 7(10)$	Aug. 2011 - Sept. 2011	Texas, USA	Forest fire (30 m.)
#3	Pleiades/WorldView2	$2000 \times 2000 \times 3(3)$	May 2012 - July 2013	Toulouse, France	Construction (0.52 m.)
#4	Spot/NDVI	$990 \times 554 \times 3(1)$	1999–2000	Gloucester, England	Flooding ($\approx$ 25m.)
#5	Radarsat-2/Google Earth	$593 \times 921 \times 1(3)$	June 2008 - Sept. 2012	Shuguang Village, China	Building construction (8 m.)
#6	Landsat-8/Sentinel-1A	$875 \times 500 \times 11(3)$	Jan. 2017 - Feb. 2017	Sutter County, USA	Flooding ($\approx$ 15m.)

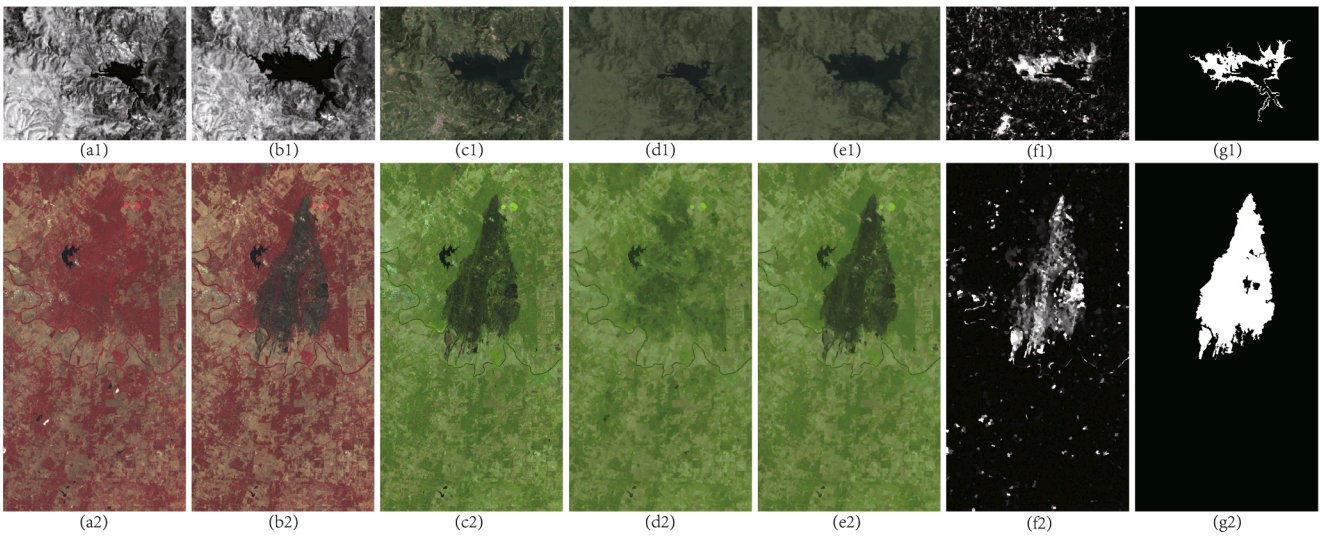


Fig. 4. Regression images of GIR-MRF on Datasets #1 and #2. In the top row, from left to right are: (a1) pre-event image $\widetilde{\mathbf{X}}_{\mathrm{NIR}}^{t1}$; (b1) post-event image $\widetilde{\mathbf{X}}_{\mathrm{NIR}}^{t2}$; (c1) post-event image $\widetilde{\mathbf{Y}}_{\mathrm{opt}}^{t2}$; (d1) regression image $\widetilde{\mathbf{Z}}_{\mathrm{NIR}}^{t1}$; (e1) regression image $\widetilde{\mathbf{X}}_{\mathrm{NIR}}^{t2}$; (f1) DI generated by GIR-MRF with $\widetilde{\mathbf{X}}_{\mathrm{NIR}}^{t1}$ and $\widetilde{\mathbf{Y}}_{\mathrm{opt}}^{t2}$; (g1) the ground truth of Dataset #1. In the bottom row, from left to right are: (a2) pre-event image $\widetilde{\mathbf{X}}_{\mathrm{L5}}^{t1}$; (b2) post-event image $\widetilde{\mathbf{X}}_{\mathrm{L5}}^{t2}$; (c2) post-event image $\widetilde{\mathbf{Y}}_{\mathrm{ALI}}^{t2}$; (d2) regression image $\widetilde{\mathbf{Z}}_{\mathrm{L5}}^{t1}$; (e2) regression image $\widetilde{\mathbf{Z}}_{\mathrm{L5}}^{t2}$; (f2) DI generated by GIR-MRF with $\widetilde{\mathbf{X}}_{\mathrm{L5}}^{t1}$ and $\widetilde{\mathbf{Y}}_{\mathrm{ALI}}^{t2}$; (g2) the ground truth of Dataset #2.

(H) GIR-MRF with the following recently proposed methods.

1) M3CD (Touati et al., 2019a): The Markov model for multimodal change detection (M3CD) is relying on an observation field built up from a pixel pairwise modeling on heterogeneous image pair, which estimates the likelihood model parameters by the standard iterative conditional estimation framework.
2) NPSG (Sun et al., 2021a): The nonlocal patch similarity graph based method (NPSG) constructs KNN graph for each image, and then compares the graphs in the same domain to measure the change level by graph mapping.
3) ALSC (Lei et al., 2020): The adaptive local structure consistency based method (ALSC) learns an adaptive KNN graph representing the local structure for each patch, and then projects this graph to the domain of the other image to detect the changes.
4) FPMS (Mignotte, 2020): The fractal projection and Markovian segmentation based method (FPMS) project the pre-event image to the domain of post-event image by fractal projection, which contains a fractal encoding step and a fractal decoding step. Then, the CM is obtained by a MRF segmentation model.
5) PSGM (Sun et al., 2020): The patch self-expression graph based method (PSGM) learns an sparse graph representing the global

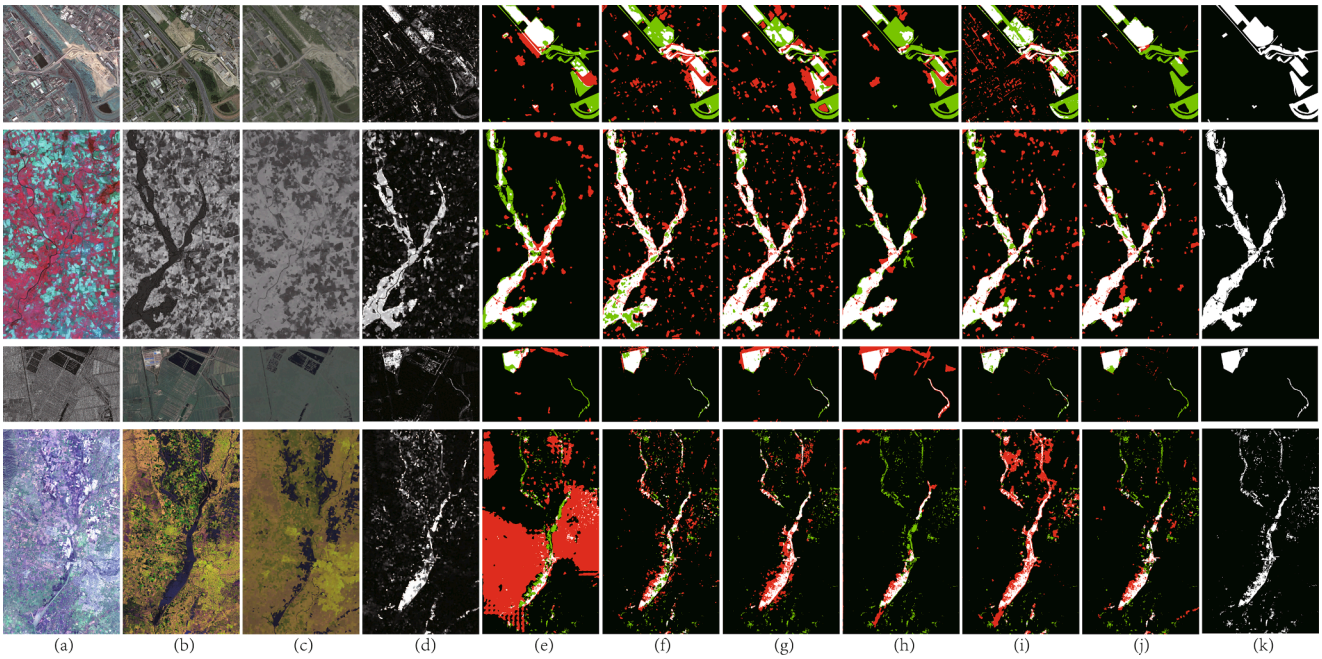


Fig. 5. DI of HGIR-MRF and binary CMs of different methods on heterogeneous data sets. From top to bottom, they correspond to Datasets #3 to #6, respectively. From left to right are: (a) pre-event image; (b) post-event image; (c) regression image of HGIR-MRF; (d) DI of HGIR-MRF; (e) binary CM of M3CD; (f) binary CM of NPSG; (g) binary CM of ALSC; (h) binary CM of FPMS; (i) binary CM of PSGM; (j) binary CM of HGIR-MRF; (k) the ground truth. In the binary CM, White: true positives (TP); Red: false positives (FP); Black: true negatives (TN); Green: false negatives (FN).

structure for each image, and then complete the image regression and calculate the DI by graph projection.

Fig. 5(c) shows the regression image of HGIR-MRF by transforming the pre-event image to the domain of post-event image. Obviously, we can see that the regression image retains the structural properties of the pre-event image, while having similar statistical properties to the post-event image, i.e., they can be directly compared. Fig. 5(d) shows the DIs of HGIR-MRF computed by (21), and Fig. 6 plots the corresponding ROC curves. As can be seen in Figs. 5(d) and 6, the DIs are able to highlight the changes very well, which demonstrates the effectiveness of Algorithm 1 and Algorithm 2 in learning the structured graph and transforming image with the structure consistency, respectively. It can also be seen from Fig. 5(d) that the generated DIs are sparse, so it is possible to obtain a satisfactory CD result by directly segmenting the DI with a simple thresholding method (such as the Otsu), which is also confirmed by the ROC curves in Fig. 6. The AUC of ROC curves on Datasets #1 to #6 are 0.892, 0.968, 0.787, 0.944, 0.970, and 0.902, respectively.

Figs. 5(e)-(j) show the binary CMs obtained by different methods. Intuitively, the CMs generated by HGIR-MRF are more consistent with the ground truth with relatively small false negatives (FN) and false positives (FP). To be specific, there are many small discontinuous error detections in NPSG, ALSC and PSGM, caused by their limited robustness to the noise and changed pixels. On the other hand, there are some continuous error detections in M3CD, especially on Dataset #6, where the method fails to detect real changes. The quantitative evaluation results of these CMs are listed in Table 7, in which the highest scores are highlighted in bold. It can be seen that the (H) GIR-MRF achieves good results on all data sets (optimal or suboptimal). At the same time, by comparing GIR-MRF and HGIR-MRF, we can find that the HGIR-MRF can yield an improvement on CD performance in general, which proves the superiority of hypergraph G^h by preserving the high-order neighborhood relations instead of pairwise ones of graph G. On the whole, the (H) GIR-MRF can suppress the false alarms and reduce the missed detections, simultaneously. This is mainly due to the following advantages of (H) GIR-MRF: 1) it incorporates both local and global information in the graph learning and image regression, which makes the structure contrast between the pre-event and post-event images more obvious and thus improves the quality of DI; 2) it uses a decomposition model in the image regression to reduce the negative impact of changed pixels, which further improves the robustness of the algorithm; 3) the change information (23) and spatial information (26) are combined in the MRF segmentation model, thus, the CM is smoother and more accurate.

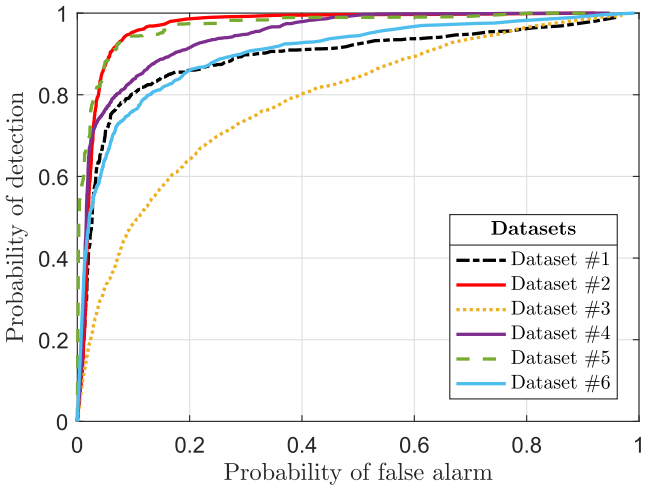


Fig. 6. ROC curves of HGIR-MRF generated DIs on different data sets.

Finally, to further compare the performance of the proposed method, the results obtained by some representative and SOTA methods (Liu et al., 2013, 2017, 2018, Luppino et al., 2019, 2021, Touati et al., 2019b, 2020; Touati and Mignotte, 2017; Zhan et al., 2018; Touati et al., 2018) are summarized in Table 8, except for M3CD (Touati et al., 2019a), NPSG (Sun et al., 2021a), ALSC (Lei et al., 2020), FPMS (Mignotte, 2020) and PSGM (Sun et al., 2020), which have been compared in Fig. 5 and Table 7. Among these comparison methods, *SCCN* (Liu et al., 2018), *LT-FL* (Zhan et al., 2018), *AFL-DSR* (Touati et al., 2020), *ACE-Net* (Luppino et al., 2021) and *X-Net* (Luppino et al., 2021) are deep learning based methods. For the sake of fairness, we directly quote the results of the corresponding datasets in their original published papers in Table 8 (because the datasets used in each paper are not identical, Table 8 is not aligned). From Table 8, we can find that the proposed (H) GIR-MRF can obtain quite competitive accuracy rate by comparing with these SOTA methods, and gain consistently good results across different data sets with an average accuracy rate of 94.5%. In addition, we believe that the application of the proposed method can be further expanded by combining it with deep learning based methods. For example, first, the proposed method can be associated with some deep learning based homogeneous CD methods after acquiring the regression image (Saha et al., 2019b, 2021b); second, the proposed method can provide assistance to some deep image translation based heterogeneous CD methods (Luppino et al., 2021; Saha et al., 2019a), such as constructing high confidence pseudo-training sets or supporting the training process.

4.4. Discussion

4.4.1. Parameter analysis

The main parameters in the (H) GIR-MRF are: the number of superpixels N_S in the superpixel segmentation process, the balancing parameter β in Algorithm 1 and Algorithm 2, the sparse regularization parameter λ in Algorithm 2, and the balancing parameter η in Algorithm 3, as listed in Table 5.

Generally, the N_S should be selected according to the image resolution and granularity requirement of CD task. A larger N_S will make the segmented superpixel smaller, which improves the detection granularity. Fig. 7 plots the regressed images and DIs generated by GIR-MRF on Dataset #6 with $N_S = 2500$, 5000, 10000 and 20000. In order to fully compare these detection results, we mark some details with the white regions in the regressed images of Fig. 7. We can find that when N_S is smaller, the size of the generated superpixels is larger, some details in the regressed image are easier to be ignored, and the block effect of the DI is more obvious as shown in Fig. 7. On the other hand, a large N_S also increases the computational complexity as analyzed in the following subsection of complexity analysis. In this paper, we simply set $N_S = 10000$, which can also be adjusted according to the task requirements and the computing environment.

The parameter β is used to balance local structure and global structure in the graph learning and image regression processes. We rewrite the global structure based penalty function (squared Frobenius norm in our experiments) as $\|\mathbf{X}-\mathbf{X}\mathbf{W}\|_F^2 = \sum_{i=1}^{N_S}\left\|\sum_{j=1}^{N_S}(\mathbf{X}_i-\mathbf{X}_j)w_{j,i}\right\|_2^2$ by using the condition of $\sum_{j=1}^{N_S} w_{j,i} = 1$. Then, we have the following inequality

Table 7
Quantitative measures of binary CMs on the heterogeneous data sets.

Methods	Dataset #3			Dataset #4			Dataset #5			Dataset #6			Average		
	OA	Kc	F1	OA	Kc	F1	OA	Kc	F1	OA	Kc	F1	OA	Kc	F1
M3CD	0.863	0.405	0.481	0.915	0.588	0.636	0.962	0.602	0.622	0.575	0.021	0.077	0.829	0.404	0.454
NPSG	0.830	0.346	0.446	0.902	0.608	0.663	0.975	0.729	0.742	0.941	0.419	0.449	0.912	0.526	0.575
ALSC	0.815	0.312	0.422	0.907	0.641	0.693	0.963	0.669	0.688	0.944	0.470	0.498	0.907	0.523	0.575
FPMS	0.838	0.215	0.296	**0.962**	**0.816**	**0.837**	0.938	0.569	0.597	0.947	0.329	0.356	0.921	0.482	0.522
PSGM	0.857	0.473	**0.558**	0.922	0.675	0.719	0.977	0.744	0.756	0.908	0.383	0.422	0.916	0.569	0.614
GIR-MRF	0.896	0.484	0.535	0.932	0.719	0.758	0.979	0.772	0.783	0.956	0.482	0.504	0.941	0.614	0.645
HGIR-MRF	**0.901**	**0.501**	0.549	0.936	0.728	0.769	**0.982**	**0.779**	**0.790**	**0.959**	**0.489**	**0.511**	**0.945**	**0.624**	**0.655**

Table 8
Accuracy rate of CMs generated by different methods on different data sets. The results of these comparison methods are reported by their original published papers. Italicized and underlined marks are used for deep learning based methods.

Dataset #3	OA	Dataset #4	OA
HGIR-MRF	**0.901**	HPT(Liu et al., 2017)	0.957–0.964
AFL-DSR(Touati et al., 2020)	0.880	**HGIR-MRF**	**0.936**
RMN(Touati et al., 2019b)	0.877	*AFL-DSR*(Touati et al., 2020)	0.836
NLPEM(Touati and Mignotte, 2017)	0.853	MDER(Liu et al., 2013)	0.818
Dataset #5	**OA**	**Dataset #6**	**OA**
HGIR-MRF	**0.982**	**HGIR-MRF**	**0.959**
AFL-DSR(Touati et al., 2020)	0.980	AMD-IR(Luppino et al., 2019)	0.933
SCCN(Liu et al., 2018)	0.976	*ACE-Net*(Luppino et al., 2021)	0.915
MDS(Touati et al., 2018)	0.967	*X-Net*(Luppino et al., 2021)	0.911
LT-FL(Zhan et al., 2018)	0.964		
RMN(Touati et al., 2019b)	0.884		

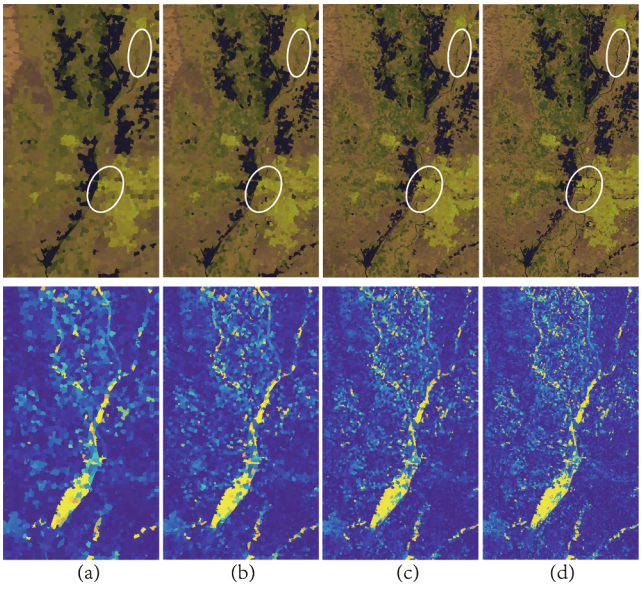


Fig. 7. Regressed images and DIs of GIR-MRF on Dataset #6 with different N_S. (a) $N_S = 2500$; (b) $N_S = 5000$; (d) $N_S = 10000$; (d) $N_S = 20000$. From (a) to (d), the corresponding AUC of DIs are 0.866, 0.881, 0.894, and 0.898, respectively.

$$\begin{aligned}
&\sum_{j=1}^{N_S}\left\|\mathbf{X}_i-\mathbf{X}_j\right\|_2^2 w_{j,i}-\left\|\sum_{j=1}^{N_S}\left(\mathbf{X}_i-\mathbf{X}_j\right)w_{j,i}\right\|_2^2\\
&=\sum_{j=1}^{N_S}\left\|\mathbf{X}_i-\mathbf{X}_j\right\|_2^2 w_{j,i}-\sum_{j=1}^{N_S}\left\|\mathbf{X}_i-\mathbf{X}_j\right\|_2^2 w_{j,i}^2-2\sum_{j=1}^{N_S}\sum_{t=j+1}^{N_S}\left(\mathbf{X}_i-\mathbf{X}_j\right)^T\left(\mathbf{X}_i-\mathbf{X}_t\right)w_{j,i}w_{t,i}\\
&=\sum_{j=1}^{N_S}\left\|\mathbf{X}_i-\mathbf{X}_j\right\|_2^2 w_{j,i}\sum_{t=1,t\neq j}^{N_S}w_{t,i}-2\sum_{j=1}^{N_S}\sum_{t=j+1}^{N_S}\left(\mathbf{X}_i-\mathbf{X}_j\right)^T\left(\mathbf{X}_i-\mathbf{X}_t\right)w_{j,i}w_{t,i}\\
&=\sum_{j=1}^{N_S}\sum_{t=j+1}^{N_S}w_{j,i}w_{t,i}\left\|\mathbf{X}_j-\mathbf{X}_t\right\|_2^2\\
&\geqslant 0,
\end{aligned} \tag{29}$$

where the second equality comes from the condition of $\sum_{j=1}^{N_S} w_{j,i} = 1$. Then, we can find that the local structure based penalty function is greater than the global structure based penalty function, i.e., $\sum_{i=1}^{N_S}\sum_{j=1}^{N_S}\left\|\mathbf{X}_i-\mathbf{X}_j\right\|_2^2 w_{j,i} \geqslant \|\mathbf{X}-\mathbf{X}\mathbf{W}\|_F^2$. Therefore, when it uses the squared Frobenius norm $g(\cdot)$, we recommend setting parameter $\beta \geqslant 1$.

The parameter λ is used to control the column-sparsity level of $\mathbf{\Delta}$ in Algorithm 2, which should be selected according to the proportion of the changed area. We demonstrate the sensitivity of our model to λ in Fig. 8, which is assessed by measuring the AUC of DI on different λ (from 10^{-6} to 1 with the ratio of 10). It illustrates that the proposed method works well over a wide range of λ.

The parameter η is used to balance the change energy term $\mathscr{J}_c(\boldsymbol{\chi})$ and the spatial energy term $\mathscr{J}_s(\boldsymbol{\chi})$ in the MRF segmentation model (22). We now fix the other parameters and change the value of η to see how it affects the CD performance. In Fig. 9, we vary $\eta/(1-\eta)$ from 10^{-2} to 10^{2} with the ratio of $10^{1/2}$. We can find that: first, with the increase of η, the

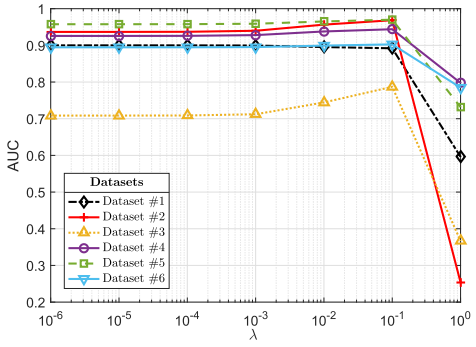


Fig. 8. Influence of parameter λ on the performance of (H) GIR-MRF.

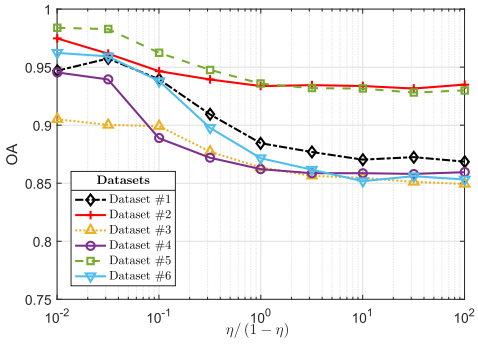


Fig. 9. Influence of parameter η on the performance of (H) GIR-MRF.

change energy term $\mathcal{J}_c(\boldsymbol{\chi})$ plays a more important role in the MRF segmentation model. Therefore, as long as the quality of DI is high enough (easy to distinguish the changed part), the final segmentation effect can be guaranteed. Second, the spatial energy term $\mathcal{J}_s(\boldsymbol{\chi})$ of (26) not only contains the spatial information of DI but also the similarity information of the original pre- and post-event images, which can be used to assist segmentation. Therefore, even if we set $\eta = 0.01$ to reduce the effect of change energy term $\mathcal{J}_c(\boldsymbol{\chi})$, (H) GIR-MRF can still obtain satisfactory results with the spatial energy term $\mathcal{J}_s(\boldsymbol{\chi})$. Fig. 10 shows the final CMs generated by HGIR-MRF with $\eta = 0.01$ and $\eta = 0.99$ on Datasets #2 and #5, from which we can see that they are all basically able to detect the changing parts, showing robustness to the parameter η.

4.4.2. Complexity analysis

The main computational complexity of the proposed (H) GIR-MRF is concentrating on the processes of structured graph learning (Algorithm 1), image regression (Algorithm 2) and MRF segmentation (Algorithm 3).

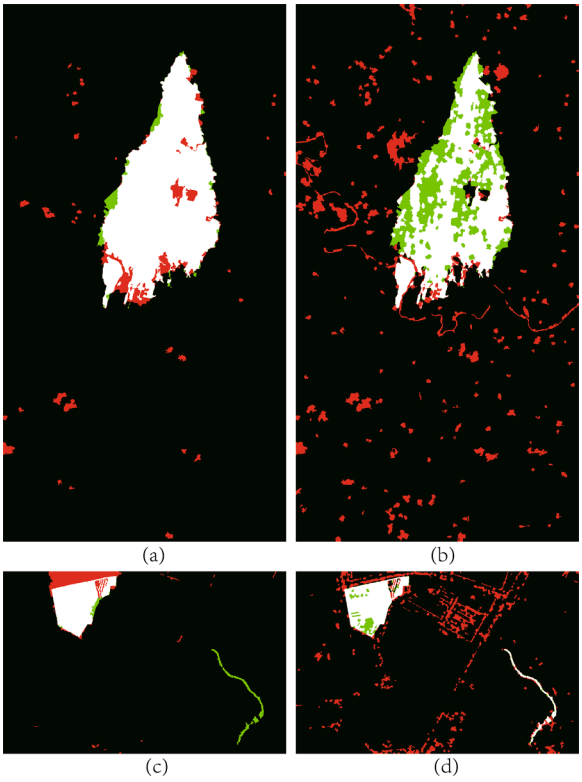


Fig. 10. Final CMs generated by HGIR-MRF with different η on Datasets #2 and #5: (a) Dataset #2 with $\eta = 0.01$; (b) Dataset #2 with $\eta = 0.99$; (c) Dataset #5 with $\eta = 0.01$; (d) Dataset #5 with $\eta = 0.99$.

Algorithm 1: first, choosing k_i for each vertex with the k selection strategy. Calculating the distances between all the superpixels requires $\mathcal{O}(3C_{\mathbf{X}}N_S^2/2)$, and sorting each distance vector to find the $k_{\max}$ nearest-neighbors of each vertex requires $\mathcal{O}(N_S \log N_S)$ by using some accelerated sorting algorithms, such as the Block sort or Tree sort. Therefore, calculating the adaptive k requires $\mathcal{O}(N_S^2 \log N_S)$. Second, updating $\boldsymbol{\varepsilon}$ with (32a). Because the proximal operation has closed-form solutions for the $g(\boldsymbol{\varepsilon})$ used in (H) GIR-MRF, updating $\boldsymbol{\varepsilon}$ requires $\mathcal{O}(3C_{\mathbf{X}}N_S^2)$ for the matrix multiplication. Third, updating $\mathbf{S}$ with (32b). The matrix inversion of $\left(\mathbf{I}_{N_S} + \frac{\mu_2}{\mu_1}\mathbf{X}^T\mathbf{X}\right)^{-1}$ requires $\mathcal{O}(N_S^3)$, the matrix multiplication for calculating $\boldsymbol{\Phi}^{(t+1)}$ requires $\mathcal{O}(3C_{\mathbf{X}}N_S^2)$, and the matrix multiplication for $\left(\mathbf{I}_{N_S} + \frac{\mu_2}{\mu_1}\mathbf{X}^T\mathbf{X}\right)^{-1}\boldsymbol{\Phi}^{(t+1)}$ requires $\mathcal{O}(N_S^3)$. Then, updating $\mathbf{S}$ requires $\mathcal{O}(N_S^3)$. Fourth, updating $\mathbf{W}$ with (32c). Because the $\mathbf{W}_i$ update with closed-form solutions requires sorting the column vector $\mathbf{P}_i$, which requires $\mathcal{O}(N_S \log N_S)$, then updating $\mathbf{W}$ needs $\mathcal{O}(N_S^2 \log N_S)$. Fifth, updating Lagrangian multipliers $\mathbf{R}_1$ and $\mathbf{R}_2$ requires $\mathcal{O}(3C_{\mathbf{X}}N_S^2)$ for the matrix multiplication.

Algorithm 2: similar to Algorithm 1, updating $\boldsymbol{\varepsilon}$ with (35a) requires $\mathcal{O}(3C_{\mathbf{Y}}N_S^2)$; updating $\boldsymbol{\Delta}$ with (35b) requires $\mathcal{O}(3C_{\mathbf{Y}}N_S)$; updating $\mathbf{Z}$ with (35c) requires $\mathcal{O}(N_S^3)$ for matrix inversion of $\boldsymbol{\Psi}^{-1}$ and $\mathcal{O}(3C_{\mathbf{Y}}N_S^2)$ for matrix multiplication of $\boldsymbol{\Theta}^{(t+1)}\boldsymbol{\Psi}^{-1}$; updating Lagrangian Multipliers $\mathbf{R}_1$ and $\mathbf{R}_2$ requires $\mathcal{O}(3C_{\mathbf{Y}}N_S^2)$ for the matrix multiplication.

Algorithm 3: step 1, assigning GMM component to each superpixel requires $\mathcal{O}\left((3C_{\mathbf{Y}})^2\mathcal{K}N_S\right)$; step 2, learning Gaussian parameters requires $\mathcal{O}(6C_{\mathbf{Y}}\mathcal{K}N_S)$; step 3, the graph-cut is completed by using min-cut/max-flow algorithm (Boykov and Kolmogorov, 2004), whose theoretical complexity and empirical complexity have been studied in Boykov and Kolmogorov (2004), that is, the theoretical complexity of the worst-case is $\mathcal{O}(2N_RN_S^2)$ with N_R representing the number of edges in the R-adjacency neighbor system. However, its empirical complexity is relatively low on typical problem instances in vision, as shown in the examples of Boykov and Kolmogorov (2004) and the following Table 9 in this paper.

Although the complexity of the proposed (H) GIR-MRF is very high in the abovementioned theoretical analysis, which requires $\mathcal{O}(N_S^3)$ for each iteration in Algorithm 1 and Algorithm 2, some acceleration strategies are available to improve the efficiency of the algorithm. First, the objective function (7) of Algorithm 1 can be written in the form of a column-wise optimization of $\mathbf{W}_i$, that is, with the columnwise independence property of $\mathbf{W}$, problem (7) can be accelerated by adopting the columnwisely parallel solution. Second, matrix inversion of $\left(\mathbf{I}_{N_S} + \frac{\mu_2}{\mu_1}\mathbf{X}^T\mathbf{X}\right)^{-1}$ in Algorithm 1 can be simplified by the Sherman–Morrison–Woodbury formula as

$$\left(\mathbf{I}_{N_S} + \frac{\mu_2}{\mu_1}\mathbf{X}^T\mathbf{X}\right)^{-1} = \mathbf{I}_{N_S} - \mathbf{X}^T\left(\frac{\mu_1}{\mu_2}\mathbf{I}_{3C_{\mathbf{X}}} + \mathbf{X}\mathbf{X}^T\right)^{-1}\mathbf{X}, \tag{30}$$

and calculated off-line in advance. Third, for the matrix inversion of $\boldsymbol{\Psi}^{-1}$ in the $\mathbf{Z}$ update of (35c), as it is fixed in the iteration framework, we can also calculate it off-line in advance. However, when N_S is very large, it may still be time-consuming to compute $\boldsymbol{\Psi}^{-1}$. Because the matrix $\boldsymbol{\Psi}$ of (36b) is a sparse, real, symmetric and positive definite matrix, the linear system of $\mathbf{Z}^{(t+1)}\boldsymbol{\Psi} = \boldsymbol{\Theta}^{(t+1)}$ can be solved efficiently by using iterative solvers, such as the conjugate gradient (CG) method. In addition, some preconditioners can also be used to accelerate CG method, such as Jacobi, incomplete Cholesky (IC), successive overrelaxation (SOR).

Table 9 reports the computational time of each process of (H) GIR-MRF with different N_S on Datasets #1 and #3. The algorithm is performed in MATLAB 2016a running on a Windows Laptop with Intel Core

Table 9
Computational time (seconds) of each process of GIR-MRF.

N_S	Dataset #1 (with size 300 × 412)					Dataset #3 (with size 2000 × 2000)				
	t_{A0}	t_{A1}	t_{A2}	t_{A3}	t_{total}	t_{A0}	t_{A1}	t_{A2}	t_{A3}	t_{total}
5000	0.61	11.32	4.41	4.15	22.65	3.01	11.68	4.63	5.05	25.76
10000	1.32	67.20	24.52	15.07	110.33	3.49	66.69	26.66	16.42	116.09
20000	2.65	428.11	155.98	34.90	625.81	4.90	451.55	181.87	36.44	679.51

i9-10980HK CPU and 64 GB of RAM. In Table 9, t_{A0} to t_{A3} represent the computational time spent in the preprocessing (superpixel segmentation and feature extraction), graph leaning (Algorithm 1), image regression (Algorithm 2) and MRF segmentation (Algorithm 3), respectively. From Table 9, we can find that: first, it is the number of superpixels rather than the size of the image that mainly determines the running time; second, Algorithms 1 and 2 are the most time-consuming processes in (H) GIR-MRF, which is in accordance with the theoretical analysis.

5. Conclusion

In this work, we proposed a structured graph learning based method to address the problem of change detection in multimodal remote sensing. In particular, it builds connections between the heterogeneous images through the inherent structure consistency. It first learns a robust graph to capture the local and global structure information of image, and then projects the graph to domain of the other image to complete the image regression, which contains a change prior based sparse constraint and two types of structure constraints: one corresponding to the global self-expression property and the other corresponding to the local similarity structure. Once the graph based image regression is performed and that a superpixel based DI is then binarized by a MRF segmentation model, which combines the change information and spatial information to improve the detection accuracy. Extensive experiments show the effectiveness of the proposed method under different CD conditions. We also hope that the proposed method will inspire the research on heterogeneous CD, especially as deep learning based methods are to be used systematically.

Due to the computational complexity, we only consider the forward transformation in this paper, i.e., translating the pre-event image to the domain of post-event image. We can also complete the backward transformation by translating the post-event image to the domain of pre-event image. Our future work is to improve the computation efficiency and design an effective fusion strategy to fuse the forward and backward transformations, thus improving the CD performance.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grant Nos. 61971426 and 4210010534. The author would like to thank the researchers for their friendly sharing of their multimodal change detection codes and data sets, which provides a wealth of resources for this study. They would also like to thank the editors and anonymous reviewers for their constructive suggestions that improved the presentation of this work.

Appendix A. Optimization of graph learning model

Problem (9) can be efficiently solved by using the alternating direction method of multipliers (ADMM). First, we introduce an auxiliary variable $\mathbf{S} \in \mathbb{R}^{N_S \times N_S}$, and rewrite the model (9) as the minimization of

$$\mathscr{L}(\boldsymbol{\varepsilon}, \mathbf{S}, \mathbf{W}, \mathbf{R}_1, \mathbf{R}_2) = Tr\left(\mathbf{W}^T \mathbf{D}^x\right) + \left\| \mathbf{W}\boldsymbol{\alpha}^{\frac{1}{2}} \right\|_F^2 + \beta g(\boldsymbol{\varepsilon}) + Tr\left(\mathbf{R}_1^T(\mathbf{W}-\mathbf{S})\right) + Tr\left(\mathbf{R}_2^T(\mathbf{X}-\mathbf{X}\mathbf{S}-\boldsymbol{\varepsilon})\right) + \frac{\mu_1}{2}\|\mathbf{W}-\mathbf{S}\|_F^2 + \frac{\mu_2}{2}\|\mathbf{X}-\mathbf{X}\mathbf{S}-\boldsymbol{\varepsilon}\|_F^2 s.t.\ \mathbf{W} \geqslant 0,\ \mathbf{W}^T\mathbf{1}_{N_S} = \mathbf{1}_{N_S}, \tag{31}$$

where $\mathbf{R}_1 \in \mathbb{R}^{N_S \times N_S}$ and $\mathbf{R}_2 \in \mathbb{R}^{3C_X \times N_S}$ are two Lagrangian multipliers, and $\mu_1, \mu_2 > 0$ are two penalty parameters. Then the alternating direction method (ADM) can be used to solve the minimization of (31) by iteratively updating one variable at a time and fixing the others. ADM separates (31) into $\boldsymbol{\varepsilon}$-subproblem, $\mathbf{S}$-subproblem and $\mathbf{W}$-subproblem. Given the current points $\left(\boldsymbol{\varepsilon}^{(t)}, \mathbf{S}^{(t)}, \mathbf{W}^{(t)}, \mathbf{R}_1^{(t)}, \mathbf{R}_2^{(t)}\right)$ at the t-th iteration, the update scheme is as followings (the detailed derivation is in the supplementary document)

$$\boldsymbol{\varepsilon}^{(t+1)} = prox_{\frac{\beta}{\mu_2} g}\left(\mathbf{X} - \mathbf{X}\mathbf{S}^{(t)} + \frac{\mathbf{R}_2^{(t)}}{\mu_2}\right), \tag{32a}$$

$$\mathbf{S}^{(t+1)} = \left(\mathbf{I}_{N_S} + \frac{\mu_2}{\mu_1}\mathbf{X}^T\mathbf{X}\right)^{-1} \boldsymbol{\Phi}^{(t+1)}, \tag{32b}$$

$$w_{(j)}^{i(t+1)} = \begin{cases} \dfrac{P_{(k_i+1)}^{i(t+1)} - P_{(j)}^{i(t+1)}}{k_i P_{(k_i+1)}^{i(t+1)} - \sum_{h=1}^{k_i} P_{(h)}^{i(t+1)}}, & j \leqslant k_i \\ 0, & j > k_i \end{cases}, \tag{32c}$$

$$\mathbf{R}_1^{(t+1)} = \mathbf{R}_1^{(t)} + \mu_1\left(\mathbf{W}^{(t+1)} - \mathbf{S}^{(t+1)}\right), \tag{32d}$$

$$\mathbf{R}_2^{(t+1)} = \mathbf{R}_2^{(t)} + \mu_2\left(\mathbf{X} - \mathbf{X}\mathbf{S}^{(t+1)} - \boldsymbol{\varepsilon}^{(t+1)}\right), \tag{32e}$$

where $\boldsymbol{\Phi}^{(t+1)} = \mathbf{W}^{(t)} + \frac{\mu_2\mathbf{X}^T\left(\mathbf{X}-\boldsymbol{\varepsilon}^{(t+1)}\right)}{\mu_1} + \frac{\mathbf{R}_1^{(t)}+\mathbf{X}^T\mathbf{R}_2^{(t)}}{\mu_1}$, $\mathbf{P}_i^{(t+1)} = \mathbf{D}_i^x + \mathbf{R}_{1i}^{(t)} - \mu_1\mathbf{S}_i^{(t+1)}$. We sort $\mathbf{P}_i^{(t+1)}$ in ascending order as $P_{(1)}^{i(t+1)}, P_{(2)}^{i(t+1)}, \cdots, P_{(N_S)}^{i(t+1)}$, and then (j) of $P_{(j)}^{i(t+1)}$ represents the position of the j-th smallest value in $\mathbf{P}_i^{(t+1)}$. The proximal operator in update of $\boldsymbol{\varepsilon}^{(t+1)}$ is defined as

$$prox_{\lambda g}(\mathbf{Y}) := \underset{\mathbf{X}}{\operatorname{argmin}} g(\mathbf{X}) + \frac{1}{2\lambda}\|\mathbf{X} - \mathbf{Y}\|_F^2. \tag{33}$$

The closed-form solutions for different $g(\cdot)$ is given in the supplementary document.

Appendix B. Optimization of image regression model

The corresponding augmented Lagrangian function of (20) is

$$\mathscr{L}(\boldsymbol{\varepsilon}, \mathbf{Z}, \boldsymbol{\Delta}, \mathbf{R}_1, \mathbf{R}_2) = 2Tr\left(\mathbf{Z}\mathbf{L}\mathbf{Z}^T\right) + \beta g(\boldsymbol{\varepsilon}) + \lambda\|\boldsymbol{\Delta}\|_{2,1} + Tr\left(\mathbf{R}_1^T(\mathbf{Z} - \mathbf{Y} - \boldsymbol{\Delta})\right) + Tr\left(\mathbf{R}_2^T(\mathbf{Z} - \mathbf{Z}\mathbf{W} - \boldsymbol{\varepsilon})\right) + \frac{\mu_1}{2}\|\mathbf{Z} - \mathbf{Y} - \boldsymbol{\Delta}\|_F^2 + \frac{\mu_2}{2}\|\mathbf{Z} - \mathbf{Z}\mathbf{W} - \boldsymbol{\varepsilon}\|_F^2, \tag{34}$$

where $\mathbf{R}_1, \mathbf{R}_2 \in \mathbb{R}^{3C_Y \times N_S}$ are two Lagrangian multipliers, and $\mu_1, \mu_2 > 0$ are two penalty parameters. Then the ADM can be used to solve the minimization of (34) by separating it into $\boldsymbol{\varepsilon}$-subproblem, $\mathbf{Z}$-subproblem and $\boldsymbol{\Delta}$-subproblem, which is similar as the procedure of solving problem (31) in Algorithm 1. Given the current points $\left(\boldsymbol{\varepsilon}^{(t)}, \mathbf{Z}^{(t)}, \boldsymbol{\Delta}^{(t)}, \mathbf{R}_1^{(t)}, \mathbf{R}_2^{(t)}\right)$ at the t-th iteration, the update scheme is

$$\boldsymbol{\varepsilon}^{(t+1)} = prox_{\frac{\beta}{\mu_2}g}\left(\mathbf{Z}^{(t)} - \mathbf{Z}^{(t)}\mathbf{W} + \frac{\mathbf{R}_2^{(t)}}{\mu_2}\right), \tag{35a}$$

$$\boldsymbol{\Delta}^{(t+1)} = prox_{\frac{\lambda}{\mu_1}\|\cdot\|_{2,1}}\left(\mathbf{Z}^{(t)} - \mathbf{Y} + \frac{\mathbf{R}_1^{(t)}}{\mu_1}\right), \tag{35b}$$

$$\mathbf{Z}^{(t+1)} = \boldsymbol{\Theta}^{(t+1)}\boldsymbol{\Psi}^{-1}, \tag{35c}$$

$$\mathbf{R}_1^{(t+1)} = \mathbf{R}_1^{(t)} + \mu_1\left(\mathbf{Z}^{(t+1)} - \mathbf{Y} - \boldsymbol{\Delta}^{(t+1)}\right), \tag{35d}$$

$$\mathbf{R}_2^{(t+1)} = \mathbf{R}_2^{(t)} + \mu_2\left(\mathbf{Z}^{(t+1)} - \mathbf{Z}^{(t+1)}\mathbf{W} - \boldsymbol{\varepsilon}^{(t+1)}\right), \tag{35e}$$

where the proximal operator in update of $\boldsymbol{\varepsilon}^{(t+1)}$ and $\boldsymbol{\Delta}^{(t+1)}$ is defined as (33), and their closed-form solutions for different $g(\cdot)$ is given in the supplementary document. The matrices of $\boldsymbol{\Theta}^{(t+1)}$ and $\boldsymbol{\Psi}$ in $\mathbf{Z}^{(t+1)}$ update is defined as

$$\boldsymbol{\Theta}^{(t+1)} = \left(\mu_2\boldsymbol{\varepsilon}^{(t+1)} - \mathbf{R}_2^{(t)}\right)(\mathbf{I}_{N_S} - \mathbf{W})^T + \mu_1\left(\mathbf{Y} + \boldsymbol{\Delta}^{(t+1)}\right) - \mathbf{R}_1^{(t)}, \tag{36a}$$

$$\boldsymbol{\Psi} = \mu_1\mathbf{I}_{N_S} + \mu_2(\mathbf{I}_{N_S} - \mathbf{W})(\mathbf{I}_{N_S} - \mathbf{W})^T + 4\mathbf{L}. \tag{36b}$$

Appendix C. Supplementary material

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.isprsjprs.2022.01.004.

References

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S., 2012. Slic superpixels compared to state-of-the-art superpixel methods. IEEE Trans. Pattern Anal. Machine Intell. 34 (11), 2274–2282.

Agarwal, S., Branson, K., Belongie, S., 2006. Higher order learning with graphs. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 17–24.

Bovolo, F., 2009. A multilevel parcel-based approach to change detection in very high resolution multitemporal images. IEEE Geosci. Remote Sens. Lett. 6 (1), 33–37.

Bovolo, F., Marchesi, S., Bruzzone, L., 2011. A framework for automatic and unsupervised detection of multiple changes in multitemporal images. IEEE Trans. Geosci. Remote Sens. 50 (6), 2196–2212.

Boykov, Y., Kolmogorov, V., 2004. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Trans. Pattern Anal. Machine Intell. 26 (9), 1124–1137.

Brunner, D., Lemoine, G., Bruzzone, L., 2010. Earthquake damage assessment of buildings using vhr optical and sar imagery. IEEE Trans. Geosci. Remote Sens. 48 (5), 2403–2420.

Chen, H., Wu, C., Du, B., Zhang, L., Wang, L., 2019. Change detection in multisource vhr images via deep siamese convolutional multiple-layers recurrent neural network. IEEE Trans. Geosci. Remote Sens. 58 (4), 2848–2864.

Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K., 2007. Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE Trans. Image Process. 16 (8), 2080–2095.

Deledalle, C.-A., Denis, L., Tupin, F., 2012. How to compare noisy patches? patch similarity beyond gaussian noise. Int. J. Comput. Vision 99 (1), 86–102.

Ebel, P., Saha, S., Zhu, X.X., 2021. Fusing multi-modal data for supervised change detection. Int. Arch. Photogramm., Remote Sens. Spatial Informat. Sci. 43, B3–2021.

Geman, S., Geman, D., 1984. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. IEEE Trans. Pattern Anal. Machine Intell. 6, 721–741.

Gong, M., Niu, X., Zhan, T., Zhang, M., 2019. A coupling translation network for change detection in heterogeneous images. Int. J. Remote Sens. 40 (9), 3647–3672.

Guan, D., Xiang, D., Tang, X., Kuang, G., 2018. Sar image despeckling based on nonlocal low-rank regularization. IEEE Trans. Geosci. Remote Sens. 57 (6), 3472–3489.

Huang, S., Kang, Z., Tsang, I.W., Xu, Z., 2019. Auto-weighted multi-view clustering via kernelized graph learning. Pattern Recogn. 88, 174–184.

Jensen, J., Ramsey, E., Mackey Jr, H., Christensen, E., Sharitz, R., 1987. Inland wetland change detection using aircraft mss data. Photogramm. Eng. Sens. 53 (5), 521–529.

Jiang, X., Li, G., Liu, Y., Zhang, X.-P., He, Y., 2020. Change detection in heterogeneous optical and sar remote sensing images via deep homogeneous feature fusion. IEEE J. Sel. Top. Appl. Earth Obser. Remote Sens. 13, 1551–1566.
Kang, Z., Peng, C., Cheng, Q., Liu, X., Peng, X., Xu, Z., Tian, L., 2021. Structured graph learning for clustering and semi-supervised classification. Pattern Recogn. 110, 107627.
Lei, L., Sun, Y., Kuang, G., 2020. Adaptive local structure consistency-based heterogeneous remote sensing change detection. IEEE Geosci. Remote Sens. Lett.
Li, H.-C., Yang, G., Yang, W., Du, Q., Emery, W.J., 2020. Deep nonsmooth nonnegative matrix factorization network with semi-supervised learning for sar image change detection. ISPRS J. Photogramm. Remote Sens. 160, 167–179 [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924271619302898.
Liu, Z.-G., Mercier, G., Dezert, J., Pan, Q., 2013. Change detection in heterogeneous remote sensing images based on multidimensional evidential reasoning. IEEE Geosci. Remote Sens. Lett. 11 (1), 168–172.
Liu, Z., Li, G., Mercier, G., He, Y., Pan, Q., 2017. Change detection in heterogenous remote sensing images via homogeneous pixel transformation. IEEE Trans. Image Process. 27 (4), 1822–1834.
Liu, J., Gong, M., Qin, K., Zhang, P., 2018. A deep convolutional coupling network for change detection based on heterogeneous optical and radar images. IEEE Trans. Neural Networks Learn. Syst. 29 (3), 545–559.
Luppino, L.T., Bianchi, F.M., Moser, G., Anfinsen, S.N., 2019. Unsupervised image regression for heterogeneous change detection. IEEE Trans. Geosci. Remote Sens. 57 (12), 9960–9975.
Luppino, L.T., Kampffmeyer, M., Bianchi, F.M., Moser, G., Serpico, S.B., Jenssen, R., Anfinsen, S.N., 2021. Deep image translation with an affinity-based change prior for unsupervised multimodal change detection. IEEE Trans. Geosci. Remote Sens.
Mercier, G., Moser, G., Serpico, S.B., 2008. Conditional copulas for change detection in heterogeneous remote sensing images. IEEE Trans. Geosci. Remote Sens. 46 (5), 1428–1441.
Mignotte, M., 2020. A fractal projection and markovian segmentation-based approach for multimodal change detection. IEEE Trans. Geosci. Remote Sens. 58 (11), 8046–8058.
Moser, G., Serpico, S.B., 2006. Generalized minimum-error thresholding for unsupervised change detection from sar amplitude imagery. IEEE Trans. Geosci. Remote Sens. 44 (10), 2972–2982.
Moser, G., Anfinsen, S.N., Luppino, L.T., Serpico, S.B., 2020. Change detection with heterogeneous remote sensing data: From semi-parametric regression to deep learning. In: IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium. IEEE, pp. 3892–3895.
Nie, F., Wang, X., Huang, H., 2014. Clustering and projected clustering with adaptive neighbors. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 977–986.
Niu, X., Gong, M., Zhan, T., Yang, Y., 2018. A conditional adversarial network for change detection in heterogeneous images. IEEE Geosci. Remote Sens. Lett. 16 (1), 45–49.
Otsu, N., 1979. A threshold selection method from gray-level histograms. IEEE Trans. Systems, Man, Cybernet. 9 (1), 62–66.
Prendes, J., Chabert, M., Pascal, F., Giros, A., Tourneret, J.-Y., 2014. A new multivariate statistical model for change detection in images acquired by homogeneous and heterogeneous sensors. IEEE Trans. Image Process. 24 (3), 799–812.
Prendes, J., Chabert, M., Pascal, F., Giros, A., Tourneret, J.-Y., 2016. A bayesian nonparametric model coupled with a markov random field for change detection in heterogeneous remote sensing images. SIAM J. Imag. Sci. 9 (4), 1889–1921.
Rother, C., Kolmogorov, V., Blake, A., 2004. Grabcut: interactive foreground extraction using iterated graph cuts. ACM Trans. Graphics (TOG) 23 (3), 309–314.
Saha, S., Bovolo, F., Bruzzone, L., 2019b. Unsupervised deep change vector analysis for multiple-change detection in vhr images. IEEE Trans. Geosci. Remote Sens. 57 (6), 3677–3693.
Saha, S., Bovolo, F., Bruzzone, L., 2019. Unsupervised multiple-change detection in vhr multisensor images via deep-learning based adaptation. In: 2019 IEEE International Geoscience and Remote Sensing Symposium, pp. 5033–5036.
Saha, S., Ebel, P., Zhu, X.X., 2021a. Self-supervised multisensor change detection. In: IEEE Trans. Geosci. Remote Sens., pp. 1–10.
Saha, S., Mou, L., Zhu, X.X., Bovolo, F., Bruzzone, L., 2021b. Semisupervised change detection using graph convolutional network. IEEE Geosci. Remote Sens. Lett. 18 (4), 607–611.
Sun, Y., Lei, L., Li, X., Sun, H., Kuang, G., 2021a. Nonlocal patch similarity based heterogeneous remote sensing change detection. Pattern Recogn. 109, 107598.
Sun, Y., Lei, L., Li, X., Tan, X., Kuang, G., 2020. Patch similarity graph matrix-based unsupervised remote sensing change detection with homogeneous and heterogeneous sensors. IEEE Trans. Geosci. Remote Sens.
Sun, Y., Lei, L., Li, X., Tan, X., Kuang, G., 2021b. Structure consistency-based graph for unsupervised change detection with homogeneous and heterogeneous remote sensing images. IEEE Trans. Geosci. Remote Sens.
Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., Rother, C., 2008. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. IEEE Trans. Pattern Anal. Machine Intell. 30 (6), 1068–1080.
Touati, R., Mignotte, M., 2017. An energy-based model encoding nonlocal pairwise pixel interactions for multisensor change detection. IEEE Trans. Geosci. Remote Sens. 56 (2), 1046–1058.
Touati, R., Mignotte, M., Dahmane, M., 2018. Change detection in heterogeneous remote sensing images based on an imaging modality-invariant mds representation. In: 2018 25th IEEE International Conference on Image Processing (ICIP). IEEE, pp. 3998–4002.
Touati, R., Mignotte, M., Dahmane, M., 2019a. Multimodal change detection in remote sensing images using an unsupervised pixel pairwise-based markov random field model. IEEE Trans. Image Process. 29, 757–767.
Touati, R., Mignotte, M., Dahmane, M., 2019b. A reliable mixed-norm-based multiresolution change detector in heterogeneous remote sensing images. IEEE J. Sel. Top. Appl. Earth Obser. Remote Sens. 12 (9), 3588–3601.
Touati, R., Mignotte, M., Dahmane, M., 2020. Anomaly feature learning for unsupervised change detection in heterogeneous images: A deep sparse residual model. IEEE J. Sel. Top. Appl. Earth Obser. Remote Sens. 13, 588–600.
Volpi, M., Camps-Valls, G., Tuia, D., 2015. Spectral alignment of multi-temporal cross-sensor images with automated kernel canonical correlation analysis. ISPRS J. Photogramm. Remote Sens. 107, 50–63.
Wan, L., Zhang, T., You, H., 2018. Multi-sensor remote sensing image change detection based on sorted histograms. Int. J. Remote Sens. 39 (11), 3753–3775.
Wan, L., Xiang, Y., You, H., 2019. An object-based hierarchical compound classification method for change detection in heterogeneous optical and sar images. IEEE Trans. Geosci. Remote Sens. 57 (12), 9941–9959.
Wang, F., Zhang, C., Li, T., 2009. Clustering with local and global regularization. IEEE Trans. Knowl. Data Eng. 21 (12), 1665–1678.
Wang, J., Yang, X., Yang, X., Jia, L., Fang, S., 2020. Unsupervised change detection between sar images based on hypergraphs. ISPRS J. Photogramm. Remote Sens. 164, 61–72.
Wu, Y., Bai, Z., Miao, Q., Ma, W., Yang, Y., Gong, M., 2020. A classified adversarial network for multi-spectral remote sensing image change detection. Remote Sens. 12 (13), 2098.
Zhan, T., Gong, M., Jiang, X., Li, S., 2018. Log-based transformation feature learning for change detection in heterogeneous images. IEEE Geosci. Remote Sens. Lett. 15 (9), 1352–1356.
Zhang, P., Gong, M., Su, L., Liu, J., Li, Z., 2016. Change detection based on deep feature representation and mapping transformation for multi-spatial-resolution remote sensing images. ISPRS J. Photogramm. Remote Sens. 116, 24–41 [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924271616000563.
Zhang, C., Fu, H., Hu, Q., Cao, X., Xie, Y., Tao, D., Xu, D., 2018. Generalized latent multi-view subspace clustering. IEEE Trans. Pattern Anal. Machine Intell. 42 (1), 86–99.
Zhang, X., Su, H., Zhang, C., Gu, X., Tan, X., Atkinson, P.M., 2021. Robust unsupervised small area change detection from sar imagery using deep learning. ISPRS J. Photogramm. Remote Sens. 173, 79–94 [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924271621000046.
Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schölkopf, B., 2004. Learning with local and global consistency. Adv. Neural Informat. Process. Syst. 16 (16), 321–328.
Zhou, D., Huang, J., Schölkopf, B., 2006. Learning with hypergraphs: Clustering, classification, and embedding. Adv. Neural Informat. Process. Syst. 19, 1601–1608.
Zhu, X., Li, X., Zhang, S., Ju, C., Wu, X., 2016. Robust joint graph sparse coding for unsupervised spectral feature selection. IEEE Trans. Neural Networks Learn. Syst. 28 (6), 1263–1275.