

Graph Signal Processing for Heterogeneous Change Detection

Yuli Sun[✉], Lin Lei[✉], Dongdong Guan[✉], Gangyao Kuang, *Senior Member, IEEE*,
and Li Liu[✉], *Senior Member, IEEE*

Abstract—This article provides a new strategy for the heterogeneous change detection (HCD) problem: solving HCD from the perspective of graph signal processing (GSP). We construct a graph to represent the structure of each image and treat each image as a graph signal defined on the graph. In this way, we convert the HCD into a GSP problem: a comparison of the responses of signals on systems defined on the graphs, which attempts to find structural differences and signal differences due to the changes between heterogeneous images. First, we analyze the GSP for HCD from the vertex domain. We show that, once a region has changed, the local structure of the image changes, i.e., the connectivity of the vertex containing this region changes. Therefore, we can compare the output signals of the same input graph signal passing through filters defined on the two graphs to detect changes. We analyze the negative effects of changing regions on the change detection results from the viewpoint of signal propagation, and we also design different filters from the vertex domain to explore the high-order neighborhood information hidden in original graphs. Second, we analyze the GSP for HCD from the spectral domain. We explore the spectral properties of different images on the same graph and show that their spectra exhibit commonalities and dissimilarities. Specifically, it is the change that leads to the dissimilarities of their spectra. With the help of graph spectral analysis, we propose a regression model for the HCD, which decomposes the source signal into the regressed signal and the changed signal, and constrains the spectral property of the regressed signal. Experiments conducted on seven real datasets show the effectiveness of the vertex domain filtering and spectral domain analysis-based HCD methods. Source code is made available at <https://github.com/yulisun/HCD-GSP>.

Index Terms—Graph signal processing (GSP), graph, heterogeneous change detection (HCD), image regression, spectral domain, structure, vertex domain.

NOMENCLATURE

$\hat{\mathbf{X}}$ and $\hat{\mathbf{Y}}$ Preevent and postevent images.
 \mathbf{X} and \mathbf{Y} Feature matrices of $\hat{\mathbf{X}}$ and $\hat{\mathbf{Y}}$.

Manuscript received 15 March 2022; revised 21 August 2022; accepted 6 November 2022. Date of publication 10 November 2022; date of current version 22 November 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFB3100800; and in part by the National Natural Science Foundation of China under Grant 61872379, Grant 12171481, and Grant 61971426. (*Corresponding author: Lin Lei*)

Yuli Sun, Lin Lei, and Gangyao Kuang are with the College of Electronic Science, National University of Defense Technology, Changsha 410073, China (e-mail: sunyuli@mail.ustc.edu.cn; alaleilin@163.com; kuangyeats@hotmail.com).

Dongdong Guan is with the High-Tech Institute of Xi'an, Xi'an 710025, China (e-mail: guandd@hotmail.com).

Li Liu is with the College of System Engineering, National University of Defense Technology, Changsha 410073, China (e-mail: li.liu@oulu.fi).

Digital Object Identifier 10.1109/TGRS.2022.3221489

$\tilde{\mathbf{X}}$ and $\check{\mathbf{X}}$	Frequency representations of \mathbf{X} .
\mathbf{X}_i	i th row of matrix \mathbf{X} .
$x_{i,j}$	i th row and j th column element of \mathbf{X} .
$G_{t1} = \{\mathcal{V}_{t1}, \mathcal{E}_{t1}, \mathbf{W}_{t1}\}$	KNN graph of the preevent image.
$G_{t2} = \{\mathcal{V}_{t2}, \mathcal{E}_{t2}, \mathbf{W}_{t2}\}$	KNN graph of the postevent image.
\mathbf{A}_{t1} and \mathbf{A}_{t2}	Adjacent matrices.
\mathbf{D}_{t1} and \mathbf{D}_{t2}	Degree matrices.
\mathbf{W}_{t1} and \mathbf{W}_{t2}	Weight matrices.
\mathbf{P}_{t1} and \mathbf{P}_{t2}	Random walk matrices.
\mathbf{L}_{t1} and \mathbf{L}_{t2}	Laplacian matrices.
\mathbf{U} and \mathbf{V}	Orthonormal matrices of eigenvectors.
Λ and Γ	Diagonal matrices of eigenvalues.
\mathbf{I}_N	$N \times N$ identity matrix.
$\mathbf{1}_N$	$N \times 1$ column vector of ones.

I. INTRODUCTION

AS A VERY important task in remote sensing, change detection (CD) aims at identifying the changes on Earth's surface by comparing multitemporal remote sensing images acquired over the same geographical area but at different times [1]. It has a wide range of real-world applications, including environmental monitoring, land management, urban development, and damage assessment [2], [3], [4]. Traditionally, most research in CD has been devoted to homogeneous CD, which performs CD with images acquired from the same sensor [5], [6], [7]. With the rapid development of high-resolution imaging techniques, increasingly more image data (such as synthetic aperture radar (SAR) images and optical images) from different types of sensors can be collected conveniently, creating even more research opportunities for detecting and monitoring subtle changes of the Earth surface at a finer scale [8], [9].

In recent years, heterogeneous CD (HCD) has emerged, starting to gain increasing attention. It addresses the problem of performing CD with images coming from different sources, such as different types of sensors (such as an SAR image and an optical image), i.e., multitemporal CD with heterogeneous remote sensing images. There are several reasons that account for this trend.

First, as each of these sensors captures different aspects and characteristics of the Earth's surface, HCD makes simultaneous use of multiple complementary data for detecting changes, overcoming the limitation of classical CD methods in relying on the availability of homogeneous data acquired by

the same sensor, which may not be available sometimes due to the environmental conditions, such as bad light or weather conditions (e.g., clouds, fogs, and rain). Second, HCD can improve the temporal resolution when monitoring long-term change evolution by inserting heterogeneous data [10]. Finally, HCD can shorten the response time of change analysis in case of sudden events (e.g., floods and earthquakes) by using the first available images to assess the damages instead of waiting for the arrival of the next homogeneous images [11].

In classical CD, we can directly compare the given images (denoted as x and y) to calculate the difference image (DI), such as via image difference in the case of optical images [12] and image ratioing (or logarithmic ratioing) in the case of SAR images [13]. However, the multitemporal images in HCD are acquired by different sensors and show quite different image characteristics [14]. The common assumption that the multitemporal images measure the same quantities and show similar statistical behavior is generally violated in HCD. As a result, most existing CD techniques do not directly apply to HCD. Therefore, the essential issue in HCD is how to make the “incomparable” images “comparable.” To achieve this, a common solution is to transform the heterogeneous images into a common domain \mathcal{Z} as $\mathcal{M}_1 : x \rightarrow z$ and $\mathcal{M}_2 : y \rightarrow z'$.

A. Related Work

With the paradigm $DI = \mathcal{M}_1(x) \ominus \mathcal{M}_2(y)$, HCD methods are proposed with different types of \mathcal{M}_1 and \mathcal{M}_2 , where \ominus denotes the difference operator. Generally, we can classify algorithms differently around the conditions, techniques, and domains of the transformation: according to whether labeled samples are required in the transformation process, HCD can be divided into supervised [15], [16], semisupervised [17], [18], and unsupervised [19], [20], [21], [22]; according to the methods used for \mathcal{M}_1 and \mathcal{M}_2 , HCD can be classified as traditional machine learning-based [15], [23], [24] and deep learning-based [25], [26], [27]; and according to the transformed common domain \mathcal{Z} , HCD can also be divided into classification comparison-based, image regression-based, and feature transformation-based.

- 1) The classification comparison-based methods first transform the images into a common category space by taking \mathcal{M}_1 and \mathcal{M}_2 as classifiers and then compare the classification results to detect changes, such as the postclassification comparison method [16], the compound classification method [28], [29], and the classified adversarial network-based method [30]. The advantages of such methods are that they are intuitive, robust to coregistration errors due to the objectwise comparison, and able to indicate the kind of change. However, classification comparison-based methods are supervised or semisupervised in order to train the accurate classifiers of \mathcal{M}_1 and \mathcal{M}_2 , and they may suffer from the risk of accumulation of classification errors.
- 2) The image regression methods first transform one image (e.g., x) to the domain of the other image (e.g., y) by setting one transformation function (\mathcal{M}_1) to the identity matrix and the other (\mathcal{M}_2) to the image translation

function and, thereby, convert the issue into the homogeneous CD. For example, some traditional methods construct the pixel-to-pixel mappings between heterogeneous images, such as the homogeneous pixel transformation (HPT) method [15], the affinity matrix-based regression method [23], and the image structure-based methods [31], [32]. Some deep translation methods have also been proposed, such as the image style transfer (IST)-based method [25], and the cycle-consistent generative adversarial network (Cycle-GAN)-based HCD methods [33], [34], [35].

- 3) The feature transformation methods transform the images into a common constructed or latent learned feature space by taking \mathcal{M}_1 and \mathcal{M}_2 as feature extraction operators. For example, some traditional methods manually construct the similarity measures that are assumed to be imaging-modality-invariant to calculate the changes, such as the copula theory-based Kullback–Leibler (KL) distance [36], the manifold learning method [24], the kernel canonical correlation analysis (kCCA) method [37], and the nonlocal pixel pairwise-based method [38]. Some deep learning methods compare the images in the latent feature spaces that are learned by the deep neural networks, such as the symmetric convolutional coupling network [39], the spatially self-paced convolutional network [40], the self-supervised learning with pseudo-Siamese networks [20], [21], the probabilistic model based on the bipartite convolutional neural network [26], the semi-supervised Siamese network [17], and the commonality autoencoder-based method [27].

Although the above methods have achieved remarkable detection results in some HCD scenes, most of them still suffer from two main challenges.

- 1) First, the connections between the heterogeneous images established by these methods are generally based on certain assumptions (e.g., some imaging-modality-invariant assumptions in traditional methods) or trained transformations (e.g., some networks in deep learning methods). These connections may be unstable and nonuniversal when the HCD scene is very complex (e.g., diversity of ground objects and difference of imaging conditions), the noise in the image is sever (especially the speckle noise in the SAR image), or the training samples are not sufficient or mixed with wrong samples.
- 2) Second, the negative influence of the unknown changed samples in the transformation is difficult to eliminate, both for training the transformation functions (\mathcal{M}_1 and \mathcal{M}_2) and completing the transformation process ($x \xrightarrow{\mathcal{M}_1} z$ and $y \xrightarrow{\mathcal{M}_2} z'$). In particular, this problem is rarely mentioned by other studies, partly for the following two reasons.
 - a) This challenge is unique to HCD, i.e., it is not a problem in the homogeneous CD that directly compares images without the design of transformations.
 - b) The previous methods usually treat HCD as a two-step process, transforming first and comparing

latter, which tends to ignore the impact of the detected changes produced by the second process on the first transformation process. In order to alleviate this negative impact, an iterative framework that combines the two processes to perform a coarse-to-fine detection is needed [41].

B. Motivations

The graph model can efficiently capture the structure information of an image, and the image processing on graphs has been proven to be effective by a large number of applications [42], [43], which has also been used in CD tasks. In the homogeneous CD of SAR images, a pointwise graph is constructed for the first image [44], and then, the DI is calculated by superimposing two images on the same graph for comparison; a pixelwise hypergraph is constructed for each image in [45], and then, the DI is calculated by matching each vertex and hyperedge between the two hypergraphs. Here, the graphs (or hypergraphs) are used to incorporate the spatial-intensity information to resist the speckle noise in SAR images.

Recently, some graph-based HCD methods have also been proposed [46]. An approximate local graph is constructed by using the Nyström extension for each image [47], and then, the graphs are fused by minimizing the similarity between the graphs to detect the changes. Based on the self-similarity property, the patchwise graphs [14], [48] and the superpixelwise graphs [31] are constructed to capture the structure of images, and then, the graphs are compared to calculate the DI by graph projection or used to perform image regression. In addition, a fractal projection method has also been proposed for HCD based on the self-similarity property [32], which projects the preevent image to the domain of postevent image with the fractal code of the preevent image.

From the above analysis, we can find that the graph-based methods have two attractive features: 1) these methods are intuitive, interpretable, and very simple, which does not require a complex training process or any labeled samples and 2) the graph can capture the inherent structural information of an image, which is robust to the noise and shows the imaging-modality-invariant property that is ideal for HCD problem. Combining with the challenges of HCD analyzed above, two aspects of the graph-based methods need to be considered in focus: first, how to adequately represent the image structure by graph; second, how to measure the structure difference in the presence of negative influence of unknown changed data. In addition, although the feasibility of these graph-based methods has been experimentally verified, they have not so far been demonstrated theoretically in depth. Furthermore, there is no unified theoretical framework for these graph-based HCD methods. These are exactly what this article is devoted to exploring.

C. Contributions

In this article, we propose a new perspective for HCD, that is, converting the HCD problem to a graph signal processing (GSP) problem. We construct a graph for each image to

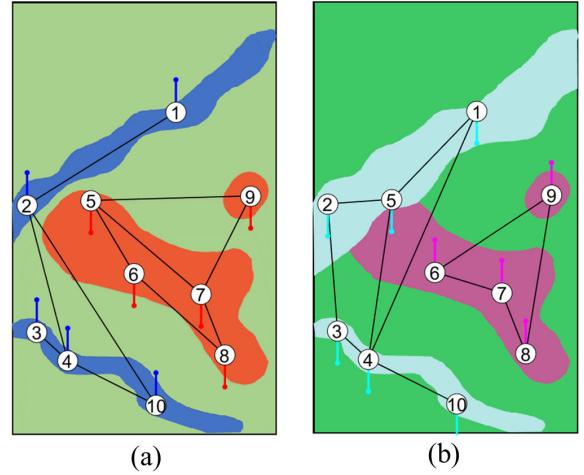


Fig. 1. Graphs and graph signals for (a) preevent image and (b) postevent image. The structure information of the image is captured by the graph topology that represents the similarity relationships between objects. The image can be treated as the signal on the graph, and the graph signal values are represented by vertical lines. Among the ten vertices in the example, the area represented by vertex 5 has changed.

capture the structure information and then treat each image as the signal on the graph, as shown in Fig. 1. In this way, the changes between heterogeneous images caused by the event will manifest themselves in two aspects: the structure difference between graphs and the signal difference on the graph, as illustrated in Fig. 2. Thereby, we can compare the responses of the two signals (i.e., images) on different systems (i.e., filters) defined on the two graphs to detect the changes from the vertex domain and the spectral domain of GSP. The main contributions of this article are given as follows.

- 1) We convert the HCD into a GSP problem and propose two frameworks to solve the HCD by employing the principles of GSP on the vertex domain and the spectral domain, respectively.
- 2) We analyze the negative influence of changes on the HCD from the perspective of signal propagation and propose a vertex domain filtering-based HCD (VDF-HCD) method that uses an iterative strategy to alleviate this influence.
- 3) We analyze the spectral properties of different images on the defined graphs and illustrate the connection between changes in heterogeneous images and differences in their spectral properties.
- 4) We propose a spectral domain analysis-based HCD (SDA-HCD) method, which decomposes the source signal into the regressed signal and the changed signal, and constrains the spectral property of the regressed signal.
- 5) We give some discussions about the proposed VDF-HCD and SDA-HCD, and show that the proposed methods can also be extended to other applications.

D. Outline

The remainder of this article is structured as follows. Section II describes the related basics of GSP. Section III describes the HCD problem from the perspective of GSP.

Section IV introduces the VDF-HCD. Section V introduces the SDA-HCD. Section VI gives some discussions of the proposed methods. Section VII presents the experimental results. Finally, Section VIII concludes this article and mentions future work. For convenience, the Nomenclature lists some important notations used in the rest of this article. One notation should be noted: \mathbf{X}_i represents the i th row of the matrix \mathbf{X} in this article, not the i th column as commonly used.

II. PRELIMINARIES

A. Vertex Filtering on Graphs

Let $G = \{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$ be a graph, and $\mathbf{f} = [f_1, \dots, f_N]^T$ be the signal on the graph G ; then, a graph shift operator \mathbf{S} is defined as *a local operation that replaces a signal value at each vertex with the linear combination of the signal values at the neighbors of that vertex* [49], [50]. Common choices for the graph shift operator are: 1) the adjacent matrix \mathbf{A} or weight matrix \mathbf{W} ; 2) the random walk (diffusion) matrix $\mathbf{P} = \mathbf{D}_w^{-1}\mathbf{W}$, where \mathbf{D}_w represents the diagonal matrix with the i th diagonal element being $\sum_{j=1}^N w_{i,j}$; and 3) the Laplacian matrix $\mathbf{L} = \mathbf{D}_w - \mathbf{W}$, symmetric normalized Laplacian matrix $\mathbf{L}^{\text{sym}} = \mathbf{D}_w^{-1/2}\mathbf{L}\mathbf{D}_w^{-1/2}$, or random-walk normalized Laplacian matrix $\mathbf{L}^{\text{rw}} = \mathbf{D}_w^{-1}\mathbf{L}$.

A linear, shift-invariant system (filter) can be defined as the polynomials in the graph shift operator \mathbf{S} of the form [51], [52], [53]

$$H(\mathbf{S}) = h_0\mathbf{S}^0 + h_1\mathbf{S} + \dots + h_M\mathbf{S}^M = \sum_{m=0}^M h_m\mathbf{S}^m \quad (1)$$

where $\mathbf{S}^0 = \mathbf{I}$, and h_0, h_1, \dots, h_M are system coefficients. The output signal of the system (1) with the input signal \mathbf{f}_{in} is

$$\mathbf{f}_{\text{out}} = H(\mathbf{S})\mathbf{f}_{\text{in}} = \sum_{m=0}^M h_m\mathbf{S}^m\mathbf{f}_{\text{in}}. \quad (2)$$

B. Spectral Filtering on Graphs

Different from the Fourier domain analysis for classic signal processing, the spectral representation of graph signals employs the eigenspectra (or simply “spectra” hereafter) of the graph shift operator \mathbf{S} given by [52]

$$\mathbf{S} = \mathbf{U}\Lambda\mathbf{U}^{-1} \quad (3)$$

where \mathbf{U} is an orthonormal matrix of the eigenvectors \mathbf{u}_k in its columns and Λ is a diagonal matrix of the corresponding eigenvalues λ_k . For the undirected graph, we have $\mathbf{S} = \mathbf{S}^T$ and $\mathbf{U}^{-1} = \mathbf{U}^T$. The graph Fourier transform (GFT) $\tilde{\mathbf{f}}$ of a graph signal \mathbf{f} is then defined as

$$\tilde{\mathbf{f}} = \text{GFT}(\mathbf{f}) = \mathbf{U}^{-1}\mathbf{f}. \quad (4)$$

The inverse GFT (IGFT) is defined as

$$\mathbf{f} = \text{IGFT}(\tilde{\mathbf{f}}) = \mathbf{U}\tilde{\mathbf{f}}. \quad (5)$$

Consider a shift-invariant system defined in (1); the output signal of \mathbf{f}_{out} of (2) can be rewritten as

$$\mathbf{f}_{\text{out}} = \sum_{m=0}^M h_m\mathbf{U}\Lambda^m\mathbf{U}^{-1}\mathbf{f}_{\text{in}} = \mathbf{U}H(\Lambda)\mathbf{U}^{-1}\mathbf{f}_{\text{in}} \quad (6)$$

where $H(\Lambda) = \sum_{m=0}^M h_m\Lambda^m$ is the transfer function of the system. Based on (6), we have $\tilde{\mathbf{f}}_{\text{out}} = \mathbf{U}^{-1}\mathbf{f}_{\text{out}} = H(\Lambda)\tilde{\mathbf{f}}_{\text{in}}$, which is the spectral domain filtering of the graph signal.

C. Frequency Ordering

In GSP, the frequency is defined by the eigenvalues of the graph shift \mathbf{S} . Specifically, we define the eigenvalue decompositions as $\mathbf{L} = \mathbf{U}\Lambda\mathbf{U}^{-1}$ and $\mathbf{W} = \mathbf{V}\Gamma\mathbf{V}^{-1}$, with $\Lambda = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_N\}$ and $\Gamma = \text{diag}\{\gamma_1, \gamma_2, \dots, \gamma_N\}$ representing the eigenvalues of \mathbf{L} and \mathbf{W} , respectively.

Definition 1 (Spectral Ordering of the Laplacian Matrix) [54], [55]: If we sort the spectra of Laplacian matrix \mathbf{L} of the graph as $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N$, then λ_N represents the lowest frequency, and λ_1 represents the highest frequency.

Definition 2 (Spectral Ordering of the Weighting Matrix) [50], [53]: If we sort the spectra of weighting matrix \mathbf{W} of the graph as $\gamma_1 \geq \gamma_2 \geq \dots \geq \gamma_N$, then γ_1 represents the lowest frequency, and γ_N represents the highest frequency.

The definitions of frequency are induced by the energy of signal change [i.e., total variation (TV)], that is, we call the frequency components with smaller variations as low frequencies and call the frequency components with higher variations as high frequencies [50]. Specifically, Definition 1 is based on the 2-Dirichlet form, i.e., a quadratic function as

$$\text{TV}_{\mathbf{L}}(\mathbf{f}) = \frac{1}{2} \sum_{(i,j) \in \mathcal{E}} w_{i,j} (\mathbf{f}_i - \mathbf{f}_j)^2 = \mathbf{f}^T \mathbf{L} \mathbf{f}. \quad (7)$$

Definition 2 is based on the 1-Dirichlet form TV on a graph with signal \mathbf{f} as

$$\text{TV}_{\mathbf{W}}(\mathbf{f}) = \|\mathbf{f} - \mathbf{W}_{\text{norm}}\mathbf{f}\|_1 \quad (8)$$

where the normalized weighting matrix $\mathbf{W}_{\text{norm}} = \mathbf{W}/\gamma_{\max}$ with $\gamma_{\max} = \max |\gamma_k|, k = 1, \dots, N$.

III. GRAPH AND HCD PROBLEM

Given two coregistered remote sensing images acquired by different sensors over the same region at different times (i.e., t_1 and t_2), denoted as $\hat{\mathbf{X}} \in \mathbb{R}^{M \times N \times C_x}$ in domain \mathcal{X} and $\hat{\mathbf{Y}} \in \mathbb{R}^{M \times N \times C_y}$ in domain \mathcal{Y} , with pixels defined as $\hat{x}(m, n, c)$ and $\hat{y}(m, n, c)$, respectively, the objective of HCD is to find the changed regions represented by a binary map ($CM \in \mathbb{R}^{M \times N}$) that labels changed and unchanged pixels.

A. Graph and Graph Signal for HCD

Since the heterogeneous images show quite different appearances and characteristics, directly comparing their pixel values is meaningless. The strategy is instead to find the connections between the topological structures of heterogeneous images. We first construct a K -nearest neighbor (KNN) graph for each image.

Definition 3 (KNN Graph): Given a set of data points $\mathbf{z} = \{z_1, z_2, \dots, z_n\}$, a KNN graph $G = \{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$ consists of n vertices connected by a set of edges \mathcal{E} and their associate weights \mathbf{W} , where $\mathcal{V} = \{1, 2, \dots, n\}$, and $(i, j) \in \mathcal{E}$ if and

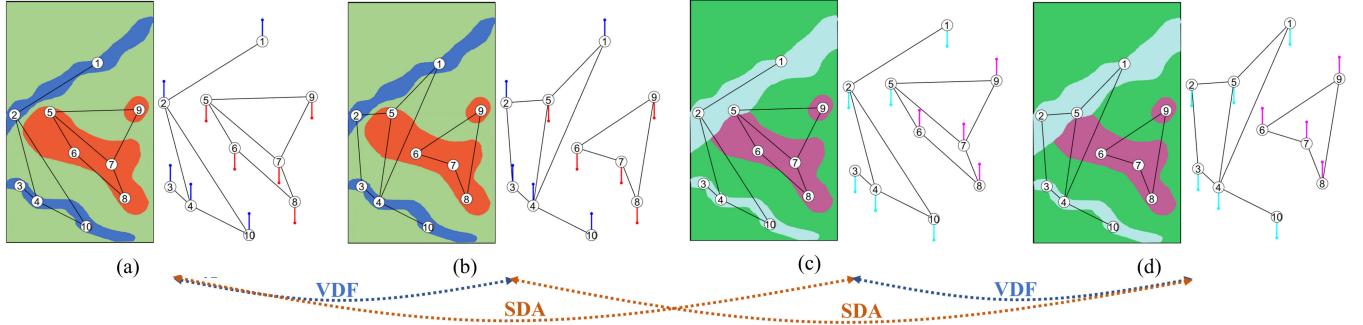


Fig. 2. Two strategies for the HCD from a GSP perspective: calculating the structure difference by letting the same graph signal pass through different graph filters (i.e., VDF-HCD) and calculating the signal difference by letting different graph signals pass through the same graph filter (i.e., SDA-HCD). (a) \mathbf{X} on G_{t1} . (b) \mathbf{X} on G_{t2} . (c) \mathbf{Y} on G_{t1} . (d) \mathbf{Y} on G_{t2} .

only if z_i is the KNN of z_j or z_j is the KNN of z_i , and $w(i, j)$ is the weight for the edge $(i, j) \in \mathcal{E}$ and zero for others.

As the scale of a pixelwise graph that sets each pixel as a vertex is very large (e.g., an image with a size of 500×500 requires 2.5×10^5 vertices), we choose the patchwise or superpixelwise KNN graph to reduce the computational cost while incorporating the contextual information for each vertex.

For the patchwise KNN graph G_{t1} of the preevent image $\hat{\mathbf{X}}$ [48], we first divide the image $\hat{\mathbf{X}}$ into a number of nonoverlapped square patches with the size $p \times p \times C_x$, and then vectorize and stack these patches into a patch group matrix $\mathbf{X} \in \mathbb{R}^{N \times M_x}$, where $M_x = p^2 C_x$ and N is the total number of patches. Then, G_{t1} can be constructed by setting each patch as a vertex with $\mathcal{V}_{t1} = \mathcal{I}$, $\mathcal{E}_{t1} = \{(i, j) | i \in \mathcal{I}; j \in \mathcal{N}_i^x\}$, where $\mathcal{I} = \{1, 2, \dots, N\}$ and $\mathcal{N}_i^x = \mathcal{N}_{\text{in}}^x \cup \mathcal{N}_{\text{out}}^x$ with

$$\begin{aligned}\mathcal{N}_{\text{in}}^x &= \{j | j \in \mathcal{I}; \mathbf{X}_i \text{ is the KNN of } \mathbf{X}_j\} \\ \mathcal{N}_{\text{out}}^x &= \{j | j \in \mathcal{I}; \mathbf{X}_j \text{ is the KNN of } \mathbf{X}_i\}.\end{aligned}\quad (9)$$

For the superpixelwise KNN graph G_{t1} [41], we first segment the images $\hat{\mathbf{X}}$ and $\hat{\mathbf{Y}}$ independently with the simple linear iterative clustering (SLIC) method [56] and then combine the segmentation maps from $\hat{\mathbf{X}}$ and $\hat{\mathbf{Y}}$ though the intersection operator to obtain the cosegmentation map $\Omega = \{\Omega_i | i \in \mathcal{I}\}$, which consists of N cosegmented superpixels of $\hat{\mathbf{X}}$ and $\hat{\mathbf{Y}}$, defined as $\hat{\mathbf{X}}_i = \{\hat{x}(m, n, c) | (m, n) \in \Omega_i, c = 1, \dots, C_x\}$ and $\hat{\mathbf{Y}}_i = \{\hat{y}(m, n, c) | (m, n) \in \Omega_i, c = 1, \dots, C_y\}$, respectively. Then, we extract M_x features (denoted as $\mathbf{X}_i \in \mathbb{R}^{M_x}$) for superpixel $\hat{\mathbf{X}}_i$ and stack these feature vectors to obtain the feature matrix $\mathbf{X} \in \mathbb{R}^{N \times M_x}$. Following that, G_{t1} can be constructed by setting each superpixel as a vertex with $\mathcal{V}_{t1} = \mathcal{I}$ and $\mathcal{E}_{t1} = \{(i, j) | i \in \mathcal{I}; j \in \mathcal{N}_i^x\}$.

For the postevent image $\hat{\mathbf{Y}}$, we can construct the patchwise or superpixelwise graph G_{t2} in a similar way as G_{t1} . Thus, the i th vertex in G_{t1} and the i th vertex in G_{t2} correspond to the same geographical location.

Once the graphs (G_{t1} and G_{t2}) are constructed to capture the structure information of the heterogeneous images, we can obtain the corresponding graph signals of $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N\}$ and $\mathbf{Y} = \{\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_N\}$.

B. HCD Problem From the Perspective of GSP

Once the graphs and graph signals are constructed, the changes between heterogeneous images can be characterized in two ways: the structure difference between the graphs of G_{t1} and G_{t2} , and the signal difference between \mathbf{X} and \mathbf{Y} on the graphs, as illustrated in Fig. 2. However, directly comparing G_{t1} , G_{t2} or \mathbf{X} , \mathbf{Y} to detect the changes is difficult, which will cause the leakage of heterogeneous data because they are constructed on different domains. Alternatively, we measure the difference by comparing the responses of the signals (i.e., \mathbf{X} and \mathbf{Y}) on systems defined on the graphs (i.e., G_{t1} and G_{t2}).

In the GSP, each vertex diffuses its information to its neighbors and also receives the information diffused from other vertices. Therefore, the GSP is a very effective tool not only for analyzing the signal but also for understanding the structure of the graph. With \mathbf{X} and \mathbf{Y} representing the graph signals, $H(\mathbf{S}_{t1})$ and $H(\mathbf{S}_{t2})$ denoting the graph filters defined on the graphs of G_{t1} and G_{t2} , respectively, we have two strategies to measure the changes between G_{t1} and G_{t2} by comparing the output signals of the same input signal on different graph filters in the vertex domain, i.e., the differences between $H(\mathbf{S}_{t1})\mathbf{X}$ and $H(\mathbf{S}_{t2})\mathbf{X}$ or $H(\mathbf{S}_{t1})\mathbf{Y}$ and $H(\mathbf{S}_{t2})\mathbf{Y}$ and 2) calculating the signal difference between \mathbf{X} and \mathbf{Y} by comparing the spectral properties of different signals on the same graph filter, i.e., the difference between $H(\mathbf{S}_{t1})\mathbf{X}$ and $H(\mathbf{S}_{t1})\mathbf{Y}$ or $H(\mathbf{S}_{t2})\mathbf{X}$ and $H(\mathbf{S}_{t2})\mathbf{Y}$ in the spectral domain, as illustrated in Fig. 2.

IV. VERTEX DOMAIN FILTERING-BASED HCD

A. Responses of Signals on Different Systems

We first consider the output of each signal on the filter defined on its own graph. Define the average weight matrix as $\mathbf{W}^{\text{avg}} \triangleq \mathbf{D}^{-1}\mathbf{A}$ with \mathbf{D} representing the degree matrix. By taking \mathbf{X} and $H(\mathbf{S}_{t1})$ as an example, with the simplest case of $H(\mathbf{S}_{t1}) = \mathbf{W}_{t1}^{\text{avg}}$, we have

$$(\mathbf{W}_{t1}^{\text{avg}}\mathbf{X})_i = \frac{1}{|\mathcal{N}_i^x|} \sum_{j \in \mathcal{N}_i^x} \mathbf{X}_j. \quad (10)$$

This filter of $(\mathbf{W}_{t1}^{\text{avg}}\mathbf{X})_i$ characterizes the concentration of information from the neighboring vertices of the understudied

vertex \mathbf{X}_i . Since \mathbf{X}_j is the KNN of \mathbf{X}_i or \mathbf{X}_i is the KNN of \mathbf{X}_j when $j \in \mathcal{N}_i^{\mathbf{x}}$, we have that \mathbf{X}_j and \mathbf{X}_i are very similar, and then, we have $(\mathbf{W}_{t1}^{\text{avg}} \mathbf{X})_i \approx \mathbf{X}_i$. This filter of $(\mathbf{W}_{t1}^{\text{avg}} \mathbf{X})_i$ can be regarded as an average smooth operator, as illustrated in Fig. 3(a) and (b).

For the case of $H(\mathbf{S}_{t1}) = \mathbf{P}_{t1}$, we have

$$(\mathbf{P}_{t1} \mathbf{X})_i = \frac{1}{\sum_{j=1}^N w_{i,j}^{t1}} \sum_{j=1}^N w_{i,j}^{t1} \mathbf{X}_j \quad (11)$$

where $w_{i,j}^{t1}$ is the (i, j) th element of \mathbf{W}_{t1} . Easily, we have $(\mathbf{P}_{t1} \mathbf{X})_i \approx \mathbf{X}_i$. This filter of $(\mathbf{P}_{t1} \mathbf{X})_i$ can be regarded as a weighted smooth operator. By comparing (10) and (11), we can find that $\mathbf{W}_{t1}^{\text{avg}} \mathbf{X}$ is a special case of $\mathbf{P}_{t1} \mathbf{X}$ due to that the adjacent matrix \mathbf{A} can be considered as a special case of the weight matrix \mathbf{W} , whereby all nonzero weights are equal to unity.

For the case of $H(\mathbf{S}_{t1}) = \mathbf{L}_{t1}^{\text{rw}}$, we have

$$(\mathbf{L}_{t1}^{\text{rw}} \mathbf{X})_i = \frac{1}{\sum_{j=1}^N w_{i,j}^{t1}} \sum_{j=1}^N w_{i,j}^{t1} (\mathbf{X}_i - \mathbf{X}_j). \quad (12)$$

Easily, we have $(\mathbf{L}_{t1}^{\text{rw}} \mathbf{X})_i = \mathbf{X}_i - (\mathbf{P}_{t1} \mathbf{X})_i \approx \mathbf{0}$. The filter of $(\mathbf{L}_{t1}^{\text{rw}} \mathbf{X})_i$ characterizes the differences between the signal value on the i th vertex and the signal values on its neighboring vertices, which can be regarded as a difference operator.

Second, we consider the output of each signal on the filter defined on the other graph and take \mathbf{X} and $H(\mathbf{S}_{t2})$ as an example. Similar to (10)–(12), we have

$$\begin{aligned} (\mathbf{W}_{t2}^{\text{avg}} \mathbf{X})_i &= \frac{1}{|\mathcal{N}_i^{\mathbf{y}}|} \sum_{j \in \mathcal{N}_i^{\mathbf{y}}} \mathbf{X}_j \\ (\mathbf{P}_{t2} \mathbf{X})_i &= \frac{1}{\sum_{j=1}^N w_{i,j}^{t2}} \sum_{j=1}^N w_{i,j}^{t2} \mathbf{X}_j \\ (\mathbf{L}_{t2}^{\text{rw}} \mathbf{X})_i &= \frac{1}{\sum_{j=1}^N w_{i,j}^{t2}} \sum_{j=1}^N w_{i,j}^{t2} (\mathbf{X}_i - \mathbf{X}_j). \end{aligned} \quad (13)$$

Third, we compare the output signals of the same signal on different graph filters. By taking \mathbf{X} and $H(\mathbf{S}) = \mathbf{W}^{\text{avg}}$ as an example, we have

$$\begin{aligned} \mathbf{d}_i^{\mathbf{x}} &= (\mathbf{W}_{t2}^{\text{avg}} \mathbf{X})_i - (\mathbf{W}_{t1}^{\text{avg}} \mathbf{X})_i \\ &= \frac{1}{|\mathcal{N}_i^{\mathbf{y}}|} \sum_{j \in \mathcal{N}_i^{\mathbf{y}}} \mathbf{X}_{j'} - \frac{1}{|\mathcal{N}_i^{\mathbf{x}}|} \sum_{j \in \mathcal{N}_i^{\mathbf{x}}} \mathbf{X}_j. \end{aligned} \quad (14)$$

Intuitively, $\mathbf{d}_i^{\mathbf{x}}$ measures the difference of signal values concentrated at the i th vertex of different systems, which is related to the local structures of two graphs at the i th vertex. Considering the signals of $(\mathbf{X}_i, \mathbf{X}_j)$ and $(\mathbf{Y}_i, \mathbf{Y}_{j'})$ connected by edges of G_{t1} and G_{t2} , respectively, and assuming that the regions represented by the j th and j' th vertices are unchanged, then we have \mathbf{X}_i and \mathbf{X}_j are very similar (representing the same kind of object), \mathbf{Y}_i and $\mathbf{Y}_{j'}$ are also very similar (representing the same kind of object), and: 1) if the i th vertex is unchanged in the event, then \mathbf{X}_j and $\mathbf{X}_{j'}$ also represent the same kind of object (showing that \mathbf{X}_j and $\mathbf{X}_{j'}$ are also very similar), which makes the elements of $\mathbf{d}_i^{\mathbf{x}}$ very small and 2) if the i th vertex is changed in the event, then \mathbf{X}_j and

$\mathbf{X}_{j'}$ represent the different kinds of object (showing that \mathbf{X}_j and $\mathbf{X}_{j'}$ are different), which makes the elements of $\mathbf{d}_i^{\mathbf{x}}$ large. Therefore, we can find that the $\mathbf{d}_i^{\mathbf{x}}$ can be used to measure the change probability (level) of the i th vertex, as illustrated in Figs. 3(a) and 4(a).

For the $H(\mathbf{S}) = \mathbf{P}$, we similarly have

$$\begin{aligned} \mathbf{d}_i^{\mathbf{x}} &= (\mathbf{P}_{t2} \mathbf{X})_i - (\mathbf{P}_{t1} \mathbf{X})_i \\ &= \sum_{j=1}^N \left(\frac{w_{i,j}^{t2} \mathbf{X}_j}{\sum_{j=1}^N w_{i,j}^{t2}} - \frac{w_{i,j}^{t1} \mathbf{X}_j}{\sum_{j=1}^N w_{i,j}^{t1}} \right). \end{aligned} \quad (15)$$

Since $\mathbf{L}^{\text{rw}} = \mathbf{I} - \mathbf{P}$, for the $H(\mathbf{S}) = \mathbf{L}^{\text{rw}}$, we have

$$\mathbf{d}_i^{\mathbf{x}} = (\mathbf{L}_{t2}^{\text{rw}} \mathbf{X})_i - (\mathbf{L}_{t1}^{\text{rw}} \mathbf{X})_i = (\mathbf{P}_{t1} \mathbf{X})_i - (\mathbf{P}_{t2} \mathbf{X})_i. \quad (16)$$

It should be noted that, when \mathbf{W} is not normalized by rows (i.e., $\mathbf{W}\mathbf{1}_N \neq c\mathbf{1}_N$ with c being a nonzero constant), $H(\mathbf{S}) = \mathbf{W}$ is not recommended for calculating the changes, i.e., $\mathbf{d}_i^{\mathbf{x}} = (\mathbf{W}_{t2} \mathbf{X})_i - (\mathbf{W}_{t1} \mathbf{X})_i$ is not appropriate with \mathbf{W} that is unnormalized by rows. This is because it will cause the leakage of heterogeneous data. For example, if we assume $\mathbf{Y} = a\mathbf{X} + b\mathbf{I}$ with a and b being nonzero constants (indicating no change between images) and set $w_{i,j}^{t1} = (1/(\|\mathbf{X}_i - \mathbf{X}_j\|_2))$ and $w_{i,j}^{t2} = (1/(\|\mathbf{Y}_i - \mathbf{Y}_j\|_2))$, then we have $\mathbf{W}_{t2} \mathbf{X} - \mathbf{W}_{t1} \mathbf{X} = ((1-a)/a)\mathbf{W}_{t1} \mathbf{X}$, which cannot indicate changes. To avoid the leakage and confusion of heterogeneous data, we use the graph shift operator of the normalized average weighting matrix \mathbf{W}^{avg} or the normalized \mathbf{P} and \mathbf{L}^{rw} for calculating the $\mathbf{d}_i^{\mathbf{x}}$ in (14)–(16).

B. Higher Order Operators

Furthermore, we only consider the simplest cases of $H(\mathbf{S}) = \mathbf{W}^{\text{avg}}, \mathbf{P}, \mathbf{L}^{\text{rw}}$ in (14)–(16). We can also choose $H(\mathbf{S})$ as the polynomials of these operators, i.e., $H(\mathbf{S}) = \sum_{m=1}^M h_m \mathbf{S}^m$. Then, we can rewrite (14)–(16) as

$$\begin{aligned} \mathbf{d}_i^{\mathbf{x}} &= (H(\mathbf{S}_{t2}) \mathbf{X})_i - (H(\mathbf{S}_{t1}) \mathbf{X})_i \\ &= \sum_{j=1}^N \sum_{m=1}^M (h_m \mathbf{S}_{t2}^m - h_m \mathbf{S}_{t1}^m) \mathbf{X}_j. \end{aligned} \quad (17)$$

Equation (17) can be interpreted in the following two ways.

- 1) $(H(\mathbf{S}_{t1}) \mathbf{X})_i = \sum_{m=1}^M h_m (\mathbf{S}_{t1}^m \mathbf{X})_i$ is a weighted sum of the attributes of vertices that are within M -hop away from the i th vertex. The coefficient h_m quantifies the contribution from the m th-hop neighbors. In this way, we can treat the $H(\mathbf{S}_{t1})$ as a new operator of a weighted higher order graph that explores the high-order neighborhood information of G_{t1} . Then, $\mathbf{d}^{\mathbf{x}} = (\sum_{m=1}^M h_m \mathbf{S}_{t2}^m - \sum_{m=1}^M h_m \mathbf{S}_{t1}^m) \mathbf{X}$ is the difference between responses of \mathbf{X} on different higher order graphs.
- 2) $\mathbf{S}_{t1}^m \mathbf{X} = \mathbf{S}_{t1} (\mathbf{S}_{t1}^{m-1} \mathbf{X})$ can be regarded as a high-order filtering process that repeats operator \mathbf{S}_{t1} by m times. Then, $\mathbf{d}^{\mathbf{x}} = \sum_{m=1}^M h_m (\mathbf{S}_{t2}^m \mathbf{X} - \mathbf{S}_{t1}^m \mathbf{X})$ is the weighted sum of differences between responses of \mathbf{X} on the filters of different orders.

Note that our aim is to discover changes in the structures between graphs (G_{t1} and G_{t2}) by comparing the responses of the same signal \mathbf{X} on different filters $H(\mathbf{S})$,

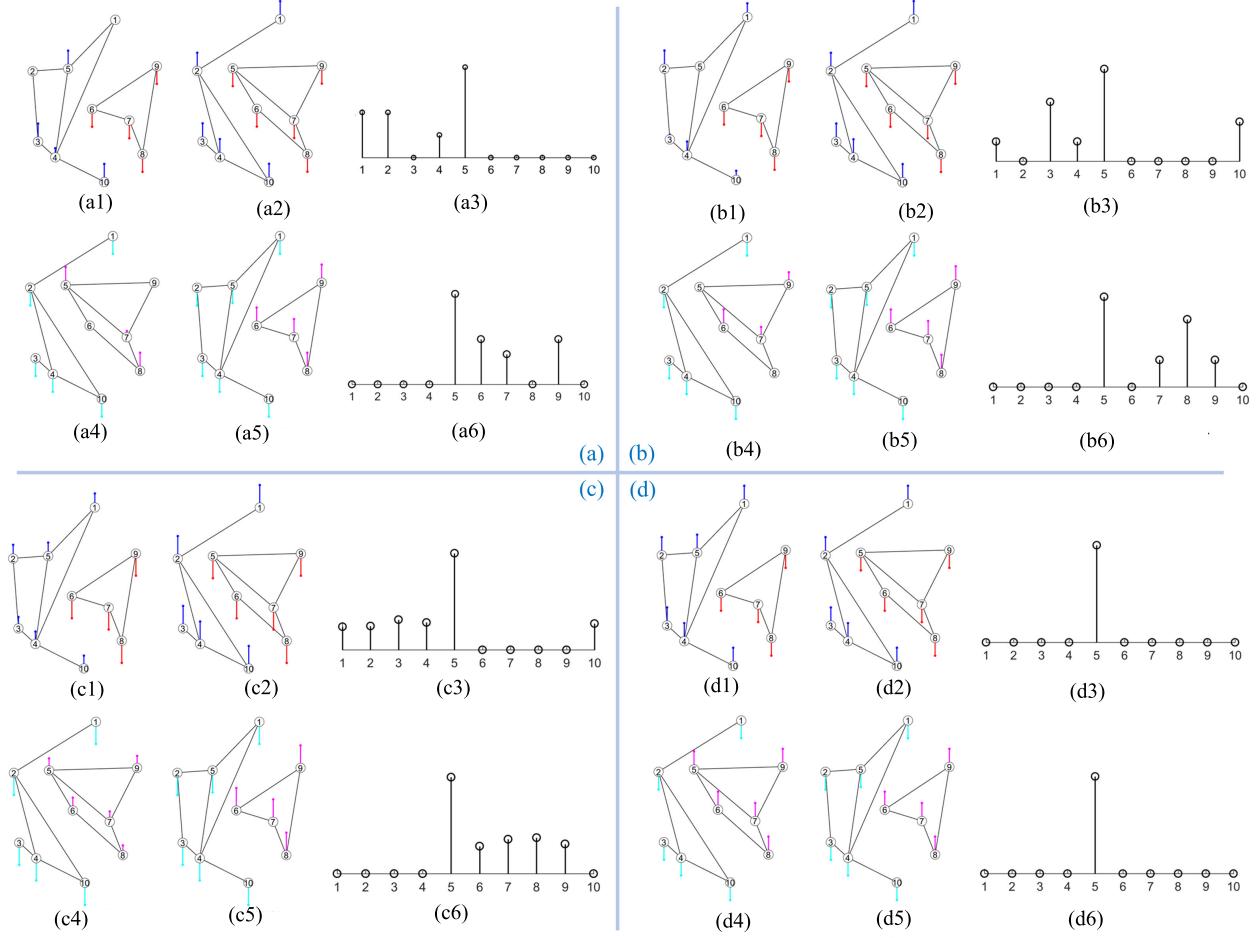


Fig. 3. Change measurements with different forms with the graphs and graph signals from Fig. 2 and the filters in Fig. 4. (a) Measure the changes using (14), where $\mathbf{W}_{t1}^{\text{avg}} \mathbf{X} = \mathbf{X}$ and $\mathbf{W}_{t2}^{\text{avg}} \mathbf{Y} = \mathbf{Y}$, $(\mathbf{W}_{t2}^{\text{avg}} \mathbf{X})_5$ differs from \mathbf{X}_5 , and $(\mathbf{W}_{t1}^{\text{avg}} \mathbf{Y})_5$ differs from \mathbf{Y}_5 due to the structure changes. (b) Measure the changes using the second-order information, where the structure changes between G_{t1} and G_{t2} can also be reflected. (c) Measure the changes using (17), where the filter $H(\mathbf{S})$ uses the form of Fig. 4(c) to make its transfer function approximate a low-pass filter. The structure difference between G_{t1} and G_{t2} shows up more clearly in Fig. 3(c) than in Fig. 3(a) and (b). (d) Measure the changes by reducing the influence of the changed vertex, where the information propagation from the changed vertex to its neighboring vertices is cut off, as illustrated by $\mathbf{W}_{t1-n}^{\text{avg}}$ and $\mathbf{W}_{t2-n}^{\text{avg}}$ in Fig. 4(d). (a1) $\mathbf{W}_{t2}^{\text{avg}} \mathbf{X}$. (a2) $\mathbf{W}_{t1}^{\text{avg}} \mathbf{X}$. (a3) $\mathbf{W}_{t2}^{\text{avg}} \mathbf{X} - \mathbf{W}_{t1}^{\text{avg}} \mathbf{X}$. (a4) $\mathbf{W}_{t1}^{\text{avg}} \mathbf{Y}$. (a5) $\mathbf{W}_{t2}^{\text{avg}} \mathbf{Y}$. (a6) $\mathbf{W}_{t1}^{\text{avg}} \mathbf{Y} - \mathbf{W}_{t2}^{\text{avg}} \mathbf{Y}$. (b1) $(\mathbf{W}_{t2}^{\text{avg}})^2 \mathbf{X}$. (b2) $(\mathbf{W}_{t1}^{\text{avg}})^2 \mathbf{X}$. (b3) $(\mathbf{W}_{t2}^{\text{avg}})^2 \mathbf{X} - (\mathbf{W}_{t1}^{\text{avg}})^2 \mathbf{X}$. (b4) $(\mathbf{W}_{t1}^{\text{avg}})^2 \mathbf{Y}$. (b5) $(\mathbf{W}_{t2}^{\text{avg}})^2 \mathbf{Y} - (\mathbf{W}_{t1}^{\text{avg}})^2 \mathbf{Y}$. (c1) $H(\mathbf{W}_{t2}^{\text{avg}}) \mathbf{X}$. (c2) $H(\mathbf{W}_{t1}^{\text{avg}}) \mathbf{X}$. (c3) $H(\mathbf{W}_{t2}^{\text{avg}}) \mathbf{X} - H(\mathbf{W}_{t1}^{\text{avg}}) \mathbf{X}$. (c4) $H(\mathbf{W}_{t1}^{\text{avg}}) \mathbf{Y}$. (c5) $H(\mathbf{W}_{t2}^{\text{avg}}) \mathbf{Y}$. (c6) $H(\mathbf{W}_{t1}^{\text{avg}}) \mathbf{Y} - H(\mathbf{W}_{t2}^{\text{avg}}) \mathbf{Y}$. (d1) $\mathbf{W}_{t2-n}^{\text{avg}} \mathbf{X}$. (d2) $\mathbf{W}_{t1}^{\text{avg}} \mathbf{X}$. (d3) $\mathbf{W}_{t2-n}^{\text{avg}} \mathbf{X} - \mathbf{W}_{t1}^{\text{avg}} \mathbf{X}$. (d4) $\mathbf{W}_{t1-n}^{\text{avg}} \mathbf{Y}$. (d5) $\mathbf{W}_{t2}^{\text{avg}} \mathbf{Y}$. (d6) $\mathbf{W}_{t1-n}^{\text{avg}} \mathbf{Y} - \mathbf{W}_{t2}^{\text{avg}} \mathbf{Y}$.

and then, the DI using this type of filters $H(\mathbf{S}) = \sum_{m=1}^M h_m \mathbf{S}^m$ is able to fully exploit information about changes in graph structures, both in terms of weighted higher order graph and weighted higher order filtering, as illustrated in Figs. 3(b) and (c) and 4(b) and (c).

C. Influence of Changes

In the analysis of (14), we have assumed that the neighboring vertex of the given i th vertex is unchanged. This assumption is reasonable due to the typical sparse prior of CD, that is, only a small part of the area changes and most areas remains unchanged during the event in practice. However, these changed vertices have a negative impact on the CD results. Next, we will show this influence from the view of signal propagation.

As shown in Fig. 3(a), for the i th vertex, if one of its KNNs in G_{t2} is changed during the event, e.g., the j 'th vertex

($j' \in \mathcal{N}_i^Y$) is changed, we have that this changed signal will propagate to the i th vertex by the graph shift operator, such as $\mathbf{W}_{t2}^{\text{avg}} \mathbf{X}$.

- 1) If the i th vertex is unchanged, this propagated signal $\mathbf{X}_{j'}$ differs significantly from \mathbf{X}_i , which increases the difference between $(\mathbf{W}_{t2}^{\text{avg}} \mathbf{X})_i$ and $(\mathbf{W}_{t1}^{\text{avg}} \mathbf{X})_i$ for the unchanged i th vertex. For example, the unchanged vertices 1, 2, and 4 connected with the changed vertex 5 in G_{t2} are affected by the changed vertex, as illustrated in Fig. 3(a3).
- 2) If the i th vertex is changed and the j 'th vertex changed to the same category as the i th vertex, then this propagated signal $\mathbf{X}_{j'}$ is quite similar as the \mathbf{X}_i , which reduces the difference between $(\mathbf{W}_{t2}^{\text{avg}} \mathbf{X})_i$ and $(\mathbf{W}_{t1}^{\text{avg}} \mathbf{X})_i$ for the changed i th vertex. From the above analysis, we can find that the changed vertex will affect the judgment of the state of other vertices whose connected neighbors

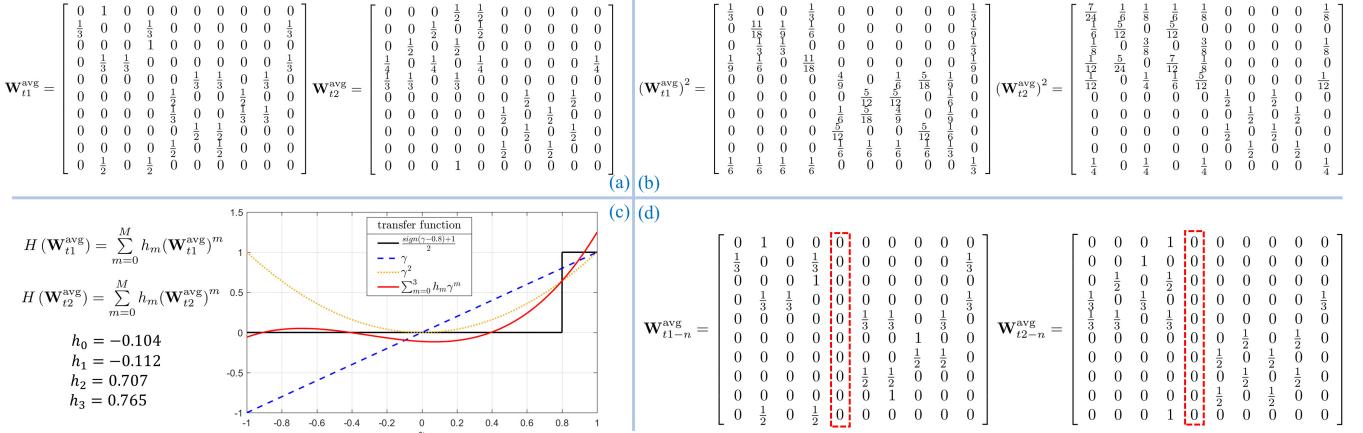


Fig. 4. Different filters used in change measurements of Fig. 3 with graphs from Fig. 2. (a) $\mathbf{W}_{t1}^{\text{avg}}$ and $\mathbf{W}_{t2}^{\text{avg}}$. (b) $(\mathbf{W}_{t1}^{\text{avg}})^2$ and $(\mathbf{W}_{t2}^{\text{avg}})^2$. (c) $H(\mathbf{W}_{t1}^{\text{avg}})$ and $H(\mathbf{W}_{t2}^{\text{avg}})$, and the transfer functions of the filters in (a)–(c). $H(\mathbf{W}_{t1}^{\text{avg}})$ and $H(\mathbf{W}_{t2}^{\text{avg}})$ are approximate low-pass filters with the transfer function $H(\Gamma) = ((\text{sign}(\Gamma - 0.8) + 1)/2)$. (d) $\mathbf{W}_{t1-n}^{\text{avg}}$ and $\mathbf{W}_{t2-n}^{\text{avg}}$, where the information propagation from the changed vertex to its neighboring vertices is cut off.

contain this change vertex, i.e., making the \mathbf{d}^x less discriminative.

D. Framework of VDF-HCD

In this section, we propose a framework for the HCD by using vertex domain filtering. We choose $H(\mathbf{S}) = \sum_{m=1}^M h_m \mathbf{P}^m$ and rewrite (17) as

$$\begin{aligned} \mathbf{d}_i^x &= (H(\mathbf{P}_{t2})\mathbf{X})_i - (H(\mathbf{P}_{t1})\mathbf{X})_i \\ &= \sum_{m=1}^M \sum_{j=1}^N h_m \left((\mathbf{P}_{t2}^m)_{i,j} \mathbf{X}_j - (\mathbf{P}_{t1}^m)_{i,j} \mathbf{X}_j \right) \\ &= \sum_{m=1}^M \sum_{j=1}^N h_m \left((\mathbf{P}_{t2}^m)_{i,j} \mathbf{X}_j - (\mathbf{P}_{t2}^m)_{i,j} \mathbf{X}_i + (\mathbf{P}_{t1}^m)_{i,j} \mathbf{X}_i - (\mathbf{P}_{t1}^m)_{i,j} \mathbf{X}_j \right) \\ &= \sum_{m=1}^M \sum_{j=1}^N h_m \left((\mathbf{P}_{t2}^m)_{i,j} - (\mathbf{P}_{t1}^m)_{i,j} \right) (\mathbf{X}_i - \mathbf{X}_j) \quad (18) \end{aligned}$$

where the second equality comes from $\mathbf{P}\mathbf{1}_N = \mathbf{1}_N$. If we choose $\mathbf{S} = \mathbf{W}^{\text{avg}}$ or $\mathbf{S} = \mathbf{L}^{\text{rw}}$, we also have

$$\mathbf{d}_i^x = \sum_{m=1}^M \sum_{j=1}^N h_m \left((\mathbf{S}_{t2}^m)_{i,j} - (\mathbf{S}_{t1}^m)_{i,j} \right) (\mathbf{X}_i - \mathbf{X}_j) \quad (19)$$

with the equations of $\mathbf{W}^{\text{avg}}\mathbf{1}_N = \mathbf{1}_N$ and $\mathbf{L}^{\text{rw}}\mathbf{1}_N = \mathbf{0}$.

Equation (19) provides the change features for each vertex, which can be regarded as the difference after the concentration of the signal difference ($\Delta_{i,j}^x = \mathbf{X}_i - \mathbf{X}_j$) at each vertex on graphs G_{t1} and G_{t2} . In order to obtain the change level of each vertex, we change the signal difference $\Delta_{i,j}^x$ as the signal distance $\text{dist}_{i,j}^x = \|\mathbf{X}_i - \mathbf{X}_j\|_2^2$ and calculate the change level as

$$\begin{aligned} f_i^x &= \sum_{m=1}^M \sum_{j=1}^N h_m \left((\mathbf{S}_{t2}^m)_{i,j} - (\mathbf{S}_{t1}^m)_{i,j} \right) \text{dist}_{i,j}^x \\ &= \sum_{j=1}^N \sum_{m=1}^M h_m \left((\mathbf{S}_{t2}^m - \mathbf{S}_{t1}^m) \odot \text{dist}^x \right)_{i,j} \quad (20) \end{aligned}$$

where dist^x represents the distance matrix of preevent image and \odot denotes the Hadamard product.

In (20), we only calculate the forward change level \mathbf{f}^x that comparing the outputs of \mathbf{X} on the graph filters of $H(\mathbf{S}_{t1})$ and $H(\mathbf{S}_{t2})$. Similarly, we can obtain the backward change level \mathbf{f}^y by comparing the outputs of \mathbf{Y} on the graph filters of $H(\mathbf{S}_{t1})$ and $H(\mathbf{S}_{t2})$, that is,

$$\mathbf{f}^y = \left(\sum_{m=1}^M h_m (\mathbf{S}_{t1}^m - \mathbf{S}_{t2}^m) \odot \text{dist}^y \right) \mathbf{1}_N \quad (21)$$

where dist^y represents the distance matrix of postevent image with element being $\text{dist}_{i,j}^y = \|\mathbf{Y}_i - \mathbf{Y}_j\|_2^2$.

In Section IV-C, we have analyzed the negative effects of the changed vertex on the change measurement from the view of signal propagation. In order to reduce this negative influence, we need to avoid propagating the signal on the changed vertex (e.g., the j th vertex) to their neighboring vertices ($\{i | i \in \mathcal{N}_j^y\}$) by setting $(\mathbf{A}_{t2})_{i,j} = 0$. However, we cannot identify, in advance, which vertices are changed, so we employ an iterative framework to complete the elimination of changed vertices in the graph construction. That is, we first calculate the change levels \mathbf{f}^x and \mathbf{f}^y and then segment them to obtain the unchanged index subset \mathcal{S} and changed index subset \mathcal{T} of vertices, which can be implemented by using some thresholding methods (such as Otsu [57]), clustering methods (such as K-means [58] and fuzzy c-means (FCMs) clustering [59]), or Markov random field (MRF)-based method [41] (adopted by VDF-HCD). After that, we propagate the unchanged vertices \mathcal{S} computed by the segmentation method of the previous round back to the graph construction process of the next round to obtain the new graphs of G_{t1-n} and G_{t2-n} , which connects each vertex with its neighbors only in the unchanged index subset \mathcal{S} . Then, we can recalculate the change levels [see (20) and (21)] as

$$\mathbf{f}^x = \left(\sum_{m=1}^M h_m (\mathbf{S}_{t2-n}^m - \mathbf{S}_{t1}^m) \odot \text{dist}^x \right) \mathbf{1}_N$$

Algorithm 1 VDF-HCD

Input: Images of $\hat{\mathbf{X}}$ and $\hat{\mathbf{Y}}$, parameters of N , $Iter$.
 Choose the operator \mathbf{S} and coefficients h_m to obtain filter $H(\mathbf{S})$.

Preprocessing: Construct the graph and graph signals
 Divide the images into patches or superpixels.
 Construct the graph signals of \mathbf{X} and \mathbf{Y} .
 Construct the KNN graphs of G_{t1} and G_{t2} .

Main iteration loop of VDF-HCD:
 Set initial index subset as $\mathcal{S}^0 = \mathcal{I}$.
 for $i = 1, 2, \dots, Iter$ do
 Construct the new graphs of G_{t1-n} and G_{t2-n} with \mathcal{S}^{i-1} .
 Calculate the change levels \mathbf{f}^x and \mathbf{f}^y by using (22).
 Segment \mathbf{f}^x and \mathbf{f}^y to obtain the \mathcal{S}^i and \mathcal{T}^i .
 end for
Output: Compute the final change map with \mathcal{S} and \mathcal{T} .

$$\mathbf{f}^y = \left(\sum_{m=1}^M h_m (\mathbf{S}_{t1-n}^m - \mathbf{S}_{t2}^m) \odot \text{dist}^y \right) \mathbf{1}_N \quad (22)$$

which can enhance the distinction between changed and unchanged classes in the change level measurement. As illustrated in Fig. 3(d), the structure difference between G_{t1} and G_{t2} shows up most clearly by using the filters in Fig. 4(d). The overall framework of the proposed VDF-HCD is summarized in Algorithm 1.

V. SPECTRAL DOMAIN ANALYSIS-BASED HCD

To avoid the leakage of heterogeneous data, in the measuring of structure difference of VDF-HCD, we weaken the effect of the difference in the original pixel values on image structure and focus more on the changes in the connectivity between vertices. Therefore, we choose the graph shift operator \mathbf{S} as the normalized average weighting matrix \mathbf{W}^{avg} (14) or the normalized random walk matrix \mathbf{P} (15) and the Laplacian matrix \mathbf{L}^{rw} (16) to measure the change level, for example, $\mathbf{d}_i^x = (\mathbf{W}_{t2}^{\text{avg}} \mathbf{X})_i - (\mathbf{W}_{t1}^{\text{avg}} \mathbf{X})_i$. Directly comparing $H(\mathbf{S}_{t1})\mathbf{X}$ and $H(\mathbf{S}_{t2})\mathbf{Y}$ will cause the leakage of heterogeneous data. However, setting $\mathbf{S} = \mathbf{W}^{\text{avg}}$ at this time does not solve the problem. Alternatively, we analyze the graph signal on the spectral domain, which can eliminate the influence of heterogeneous data from different domains.

Nevertheless, spectral domain analysis also presents another challenge: how to get the change of signal from the difference of spectral domain. Since the goal of HCD is to find the region where the change occurred during the event, it is directly corresponding to the region represented by the vertex in the graph. Therefore, the change measurement of $\mathbf{d}_i^x = (\mathbf{W}_{t2}^{\text{avg}} \mathbf{X})_i - (\mathbf{W}_{t1}^{\text{avg}} \mathbf{X})_i$ in VDF-HCD that finds the changes in the vertex domain can directly output the HCD results: the changed vertex. As we know, in classical signal processing, the Fourier coefficients are the integration or summation form of the signal in the time domain. Similarly, the graph frequency coefficients are also the summation form of the graph signal in the vertex domain. Therefore, directly comparing the spectra

of two graph signals in the spectral domain is not possible to find the changed vertex (i.e., changed region).

A. Signal Decomposition

To address the challenge introduced by the spectral analysis, we use a signal decomposition method: we decompose the source signal \mathbf{Y} into the regressed signal \mathbf{Z} and the changed signal Δ as $\mathbf{Y} = \mathbf{Z} + \Delta$, and require the regressed signal \mathbf{Z} to have the same spectral property as the target signal \mathbf{X} on the graph G_{t1} . With this decomposition model, we can obtain the changed signal Δ , which can be directly used to detect the areas of change. At the same time, since the signal decomposition model directly outputs both the regressed signal and changed signal, which means that the changed samples are removed from the original \mathbf{Y} , the negative influence of the changed samples on the regression process is reduced. On the other hand, if we treat $\hat{\mathbf{Z}}$ as the regression image, because \mathbf{Z} and \mathbf{X} have the same spectral property, the structures of $\hat{\mathbf{Z}}$ and $\hat{\mathbf{X}}$ are consistent. Because the regressed signal \mathbf{Z} is separated from \mathbf{Y} , then $\hat{\mathbf{Z}}$ and $\hat{\mathbf{Y}}$ belong to the same domain. Therefore, this signal decomposition model can also be regarded as an image regression method.

We mainly consider the case of forward regression that transforms the preevent image to the domain of the postevent image with $\mathbf{Y} = \mathbf{Z} + \Delta$ and do not consider the backward regression of transforming the postevent image to the domain of preevent image, which is a similar process to the former. Therefore, for the sake of simplicity in defining notation, we use the eigenvalue decompositions of \mathbf{L}_{t1} and \mathbf{W}_{t1} in the latter part of this article as $\mathbf{L}_{t1} = \mathbf{U}\Lambda\mathbf{U}^{-1}$ and $\mathbf{W}_{t1} = \mathbf{V}\Gamma\mathbf{V}^{-1}$ in descending order of eigenvalues, with $\Lambda = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_N\}$, $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N$, and $\Gamma = \text{diag}\{\gamma_1, \gamma_2, \dots, \gamma_N\}$, $\gamma_1 \geq \gamma_2 \geq \dots \geq \gamma_N$, respectively. Next, we first analyze the spectral properties of the heterogeneous images on different graphs and then propose an HCD method based on the spectral domain analysis.

B. Low-Pass Property

As the frequency can be indicated by the rate of change between the vertices on the edges as illustrated by $\text{TV}_{\mathbf{L}}$ (7) and $\text{TV}_{\mathbf{W}}$ (8), we can intuitively conclude that \mathbf{X} is the low-pass signal on G_{t1} , that is, the TVs of $\text{TV}_{\mathbf{L}_{t1}}(\mathbf{X})$ and $\text{TV}_{\mathbf{W}_{t1}}(\mathbf{X})$ are very small. This can be observed in the construction process of the KNN graph, where the i th vertex and the j th vertex are connected if and only if $i \in \mathcal{N}_j^x$ (i.e., \mathbf{X}_i belongs to the KNN of \mathbf{X}_j or \mathbf{X}_j belongs to the KNN of \mathbf{X}_i), which means that \mathbf{X}_i and \mathbf{X}_j are very similar.

Remark 1: The graph signal \mathbf{X} is an approximate low-pass signal on the KNN graph G_{t1} , and the graph signal \mathbf{Y} is an approximate low-pass signal on the KNN graph G_{t2} .

Specifically, Remark 1 can also be demonstrated by using the Definitions 1 and 2, that is, the high-frequency component of $\hat{\mathbf{X}}$ is almost zero. Substituting \mathbf{L}_{t1} and \mathbf{X} into (7), we have

$$\begin{aligned} \text{TV}_{\mathbf{L}_{t1}}(\mathbf{X}) &= \text{Tr}(\mathbf{X}^T \mathbf{L}_{t1} \mathbf{X}) = \text{Tr}(\mathbf{X}^T \mathbf{U} \Lambda \mathbf{U}^{-1} \mathbf{X}) \\ &= \text{Tr}(\tilde{\mathbf{X}}^T \Lambda \tilde{\mathbf{X}}) = \sum_{k=1}^N \lambda_k \|\tilde{\mathbf{X}}_k\|_2^2 \end{aligned} \quad (23)$$

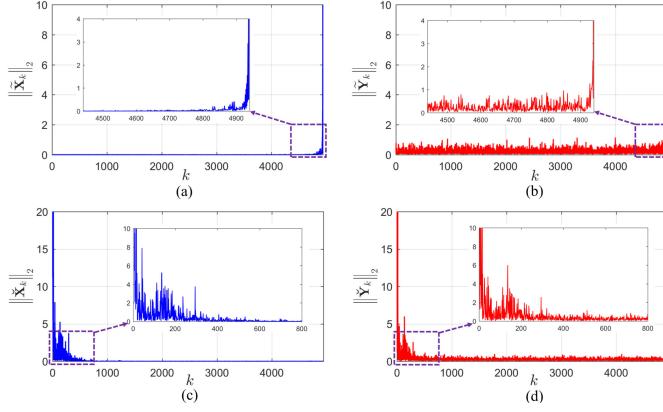


Fig. 5. Spectral properties of graph signals \mathbf{X} and \mathbf{Y} on the KNN graph G_{t1} , where \mathbf{X} , \mathbf{Y} , and G_{t1} are constructed from the preevent and postevent images in Fig. 6. (a) $\|\tilde{\mathbf{X}}_k\|_2$ with $\tilde{\mathbf{X}} = \mathbf{U}^{-1}\mathbf{X}$. (b) $\|\tilde{\mathbf{Y}}_k\|_2$ with $\tilde{\mathbf{Y}} = \mathbf{U}^{-1}\mathbf{Y}$. (c) $\|\check{\mathbf{X}}_k\|_2$ with $\check{\mathbf{X}} = \mathbf{V}^{-1}\mathbf{X}$. (d) $\|\check{\mathbf{Y}}_k\|_2$ with $\check{\mathbf{Y}} = \mathbf{V}^{-1}\mathbf{Y}$.

where $\tilde{\mathbf{X}} = \mathbf{U}^{-1}\mathbf{X}$. Because $TV_{L_{t1}}(\mathbf{X}) = (1/2)\sum_{(i,j) \in \mathcal{E}_{t1}} w_{i,j}^{t1} \text{dist}_{i,j}^{\mathbf{X}}$ is very small, then it requires that $\|\tilde{\mathbf{X}}_k\|_2^2$ corresponding to the large λ_k should also be very small. According to Definition 1 that large λ_k represents the high frequency, we have that the high-frequency component in $\tilde{\mathbf{X}}$ is very small.

We define the normalized symmetrized weighting matrix as

$$\mathbf{W}^{\text{sym}} = \Phi^{-1/2} \mathbf{W} \Phi^{-1/2} \quad (24)$$

where the diagonal matrix Φ is defined such that $\mathbf{W}^{\text{sym}} \mathbf{1} = \mathbf{1}$ and $(\mathbf{W}^{\text{sym}})^T \mathbf{1} = \mathbf{1}$. \mathbf{W}^{sym} is a symmetric doubly stochastic matrix, and it can be obtained by applying the Sinkhorn–Knopp balancing algorithm [60], [61], [62] to iteratively normalizes the rows and columns of \mathbf{W} until convergence. For the weighting matrix \mathbf{W}^{sym} , we have $1 = \gamma_1 \geq \gamma_2 \geq \dots \geq \gamma_N \geq 0$, which follows from the property of symmetric doubly stochastic matrix. We have

$$\begin{aligned} \|\mathbf{W}_{t1}^{\text{sym}} \mathbf{X}\|_F^2 &= \|\mathbf{V} \Gamma \mathbf{V}^{-1} \mathbf{X}\|_F^2 = \text{Tr}(\mathbf{X}^T \mathbf{V} \Gamma^2 \mathbf{V}^{-1} \mathbf{X}) \\ &= \sum_{k=1}^N \gamma_k^2 \|\check{\mathbf{X}}_k\|_2^2 \end{aligned} \quad (25)$$

where $\check{\mathbf{X}} = \mathbf{V}^{-1}\mathbf{X}$. Due to the conservation of energy, we also have $\|\mathbf{X}\|_F^2 = \|\check{\mathbf{X}}\|_F^2 = \sum_{k=1}^N \|\check{\mathbf{X}}_k\|_2^2$. As $\mathbf{W}_{t1}^{\text{sym}} \mathbf{X} \approx \mathbf{X}$ and it requires $\sum_{k=1}^N (1 - \gamma_k^2) \|\check{\mathbf{X}}_k\|_2^2 \approx 0$, then we have $\|\check{\mathbf{X}}_k\|_2^2$ corresponding to the small γ_k is also very small. According to the Definitions 1 and 2 that small γ_k represents the high frequency, we have that the high-frequency component is almost zero in $\check{\mathbf{X}}$. Fig. 5(a) and (c) shows the illustration of this Remark 1.

C. High-Pass Property

Next, we consider the spectral property of \mathbf{Y} on the graph G_{t1} . As shown in Fig. 5, we find that, different from the $\tilde{\mathbf{X}}$ (or $\check{\mathbf{X}}$) on G_{t1} , the high-frequency component of $\tilde{\mathbf{Y}}$ (or $\check{\mathbf{Y}}$) on G_{t1} is not equal to zero, which seems like the noisy component in the classic signal processing. We decompose \mathbf{Y} into the regressed signal \mathbf{Z} and the changed signal Δ as $\mathbf{Y} = \mathbf{Z} + \Delta$, where \mathbf{Z} is the assumed unchanged signal that represents the translated \mathbf{X} in the domain of \mathbf{Y} .

Remark 2: The regressed signal \mathbf{Z} is an approximate low-pass signal on the KNN graph G_{t1} . The high-frequency component of $\tilde{\mathbf{Y}}$ (or $\check{\mathbf{Y}}$) on G_{t1} is introduced by the changes caused by the event.

First, we have that the structure of \mathbf{Z} is as same as \mathbf{X} , that is, if \mathbf{X}_i and \mathbf{X}_j represent the same kind (or different kinds) of the object, then \mathbf{Z}_i and \mathbf{Z}_j also represent the same kind (or different kinds) of the object. Then, the similarity relationships of \mathbf{Z}_i and \mathbf{Z}_j are the same as \mathbf{X}_i and \mathbf{X}_j . Therefore, based on the definition of frequency or by using the $TV_{L_{t1}}(\mathbf{Z})$ or $TV_{W_{t1}}(\mathbf{Z})$, we have that the regressed \mathbf{Z} has the same spectral property as \mathbf{X} on the graph G_{t1} : both \mathbf{Z} and \mathbf{X} are the approximate low-pass signals on G_{t1} .

Second, taking the GFT of \mathbf{Y} on G_{t1} with the Laplacian matrix \mathbf{L}_{t1} , we have

$$\tilde{\mathbf{Y}} = \text{GFT}(\mathbf{Y}) = \mathbf{U}^{-1}(\mathbf{Z} + \Delta) = \tilde{\mathbf{Z}} + \tilde{\Delta}. \quad (26)$$

Because the high-frequency component of $\tilde{\mathbf{Z}}$ is approximately equal to zero, we have $\tilde{\mathbf{Y}}_k \approx \tilde{\Delta}_k$ for high frequency λ_k , that is, the high-frequency component of $\tilde{\mathbf{Y}}$ in graph G_{t1} is introduced by the changes Δ caused by the event, as illustrated in Fig. 5(b) and (d). Meanwhile, it should be noted that the low-frequency component of $\tilde{\Delta}_k$ is not always equal to zero. For example, there will also be cases where adjacent vertices change at the same time, as shown in Fig. 6(c).

D. Regression With the Low-Frequency Components

If we divide $\tilde{\mathbf{Y}}$ into the low- and high-frequency parts as $\tilde{\mathbf{Y}} = \tilde{\mathbf{Y}}_L + \tilde{\mathbf{Y}}_H$ by using an ideal low-pass filter with cutoff eigenvalue λ_{k_c} , i.e.,

$$\begin{aligned} (\tilde{\mathbf{Y}}_L)_k &= \begin{cases} \tilde{\mathbf{Y}}_k; & \text{if } k \geq k_c \\ 0; & \text{else} \end{cases} \\ (\tilde{\mathbf{Y}}_H)_k &= \begin{cases} \tilde{\mathbf{Y}}_k; & \text{if } k < k_c \\ 0; & \text{else.} \end{cases} \end{aligned} \quad (27)$$

With this low-pass filter, we can also divide $\tilde{\mathbf{Z}}$ and $\tilde{\Delta}$ as $\tilde{\mathbf{Z}} = \tilde{\mathbf{Z}}_L + \tilde{\mathbf{Z}}_H$ and $\tilde{\Delta} = \tilde{\Delta}_L + \tilde{\Delta}_H$, respectively. Based on the $\tilde{\mathbf{Y}}_H \approx \tilde{\Delta}_H$, we have

$$\begin{aligned} \mathbf{Z} &= \text{IGFT}(\tilde{\mathbf{Z}}) \approx \mathbf{U} \tilde{\mathbf{Y}}_L - \mathbf{U} \tilde{\Delta}_L \\ \Delta &= \text{IGFT}(\tilde{\Delta}) \approx \mathbf{U} \tilde{\mathbf{Y}}_H + \mathbf{U} \tilde{\Delta}_L. \end{aligned} \quad (28)$$

If we ignore the low-frequency component of $\tilde{\Delta}_L$, we can obtain an approximate \mathbf{Z}' and Δ' as $\mathbf{Z}' = \mathbf{U} \tilde{\mathbf{Y}}_L$ and $\Delta' = \mathbf{U} \tilde{\mathbf{Y}}_H$. Fig. 6 shows the approximate regression image and the changed image.

E. Regression Model of SDA-HCD

From the decomposition model $\mathbf{Y} = \mathbf{Z} + \Delta$, we have that it requires the regressed \mathbf{Z} to have the same spectral property (i.e., low-pass) as \mathbf{X} in the KNN graph G_{t1} , which can be used as a constraint for the \mathbf{Z} . On the other hand, for the changed signal Δ , we have a change prior-based sparse constraint for the CD problem, which is based on the fact that only a small part of the area changed and most areas remain unchanged during the event in practice. Therefore, we can

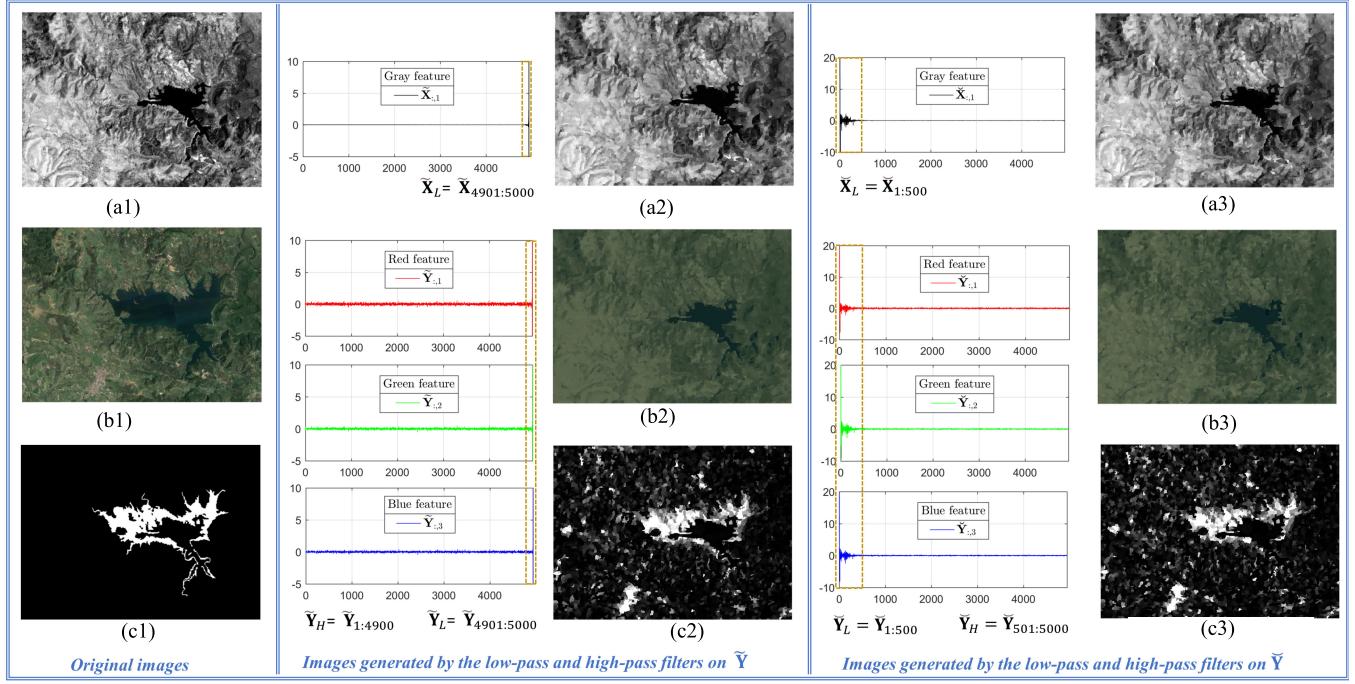


Fig. 6. Regression and changed images generated by the low- and high-pass filters defined on the KNN graph G_{t1} , respectively. By comparing (a2) and (a3) with (a1), we can find that the original $\hat{\mathbf{X}}$ is very similar to the reconstructed $\hat{\mathbf{X}}'$ and $\hat{\mathbf{X}}''$ with the low-frequency components of $\hat{\mathbf{X}}_L$ and $\check{\mathbf{X}}_L$, respectively. For the regressed $\hat{\mathbf{Y}}'$ (b2) and $\hat{\mathbf{Y}}''$ (b3) with $\hat{\mathbf{Y}}_L$ and $\check{\mathbf{Y}}_L$, it can be found that their structures are consistent with the preevent image; however, the appearances are similar to that of the postevent image. For the generated DIs of (c2) and (c3), we can see that the high-frequency components of $\hat{\mathbf{Y}}_H$ and $\check{\mathbf{Y}}_H$ can be used to detect the changes. (a1) Pre-event image $\hat{\mathbf{X}}$. (a2) $\hat{\mathbf{X}}'$ with $\mathbf{X}' = \mathbf{U}\hat{\mathbf{X}}_L$. (a3) $\hat{\mathbf{X}}''$ with $\mathbf{X}'' = \mathbf{V}\check{\mathbf{X}}_L$. (b1) Pre-event image $\hat{\mathbf{Y}}$. (b2) $\hat{\mathbf{Y}}'$ with $\mathbf{Y}' = \mathbf{U}\hat{\mathbf{Y}}_L$. (b3) $\hat{\mathbf{Y}}''$ with $\mathbf{Y}'' = \mathbf{V}\check{\mathbf{Y}}_L$. (c1) Change map. (c2) $\hat{\mathbf{D}}\mathbf{I}'$ with $\mathbf{Z}' = \mathbf{U}\hat{\mathbf{Y}}_H$. (c3) $\hat{\mathbf{D}}\mathbf{I}''$ with $\mathbf{Z}'' = \mathbf{V}\check{\mathbf{Y}}_H$.

obtain a decomposition-based regression model for the HCD problem

$$\min_{\mathbf{Z}, \Delta} g(\mathbf{Z}) + \alpha f(\Delta) \quad \text{s.t. } \mathbf{Y} = \mathbf{Z} + \Delta \quad (29)$$

where $g(\mathbf{Z})$ represents the spectral constraint for \mathbf{Z} , $f(\Delta)$ represents the prior regularization for Δ , and $\alpha > 0$ is a balancing parameter.

First, for the spectral constraint $g(\mathbf{Z})$, we have different choices. For example, we can constrain $\tilde{\mathbf{Z}}_H \approx \mathbf{0}$ by using the penalties: 1) $g(\mathbf{Z}) = \|\mathbf{U}_H^{-1}\mathbf{Z}\|_F^2$, where $\mathbf{U}_H^{-1} = (\mathbf{U}^{-1})_{1:k_c,:}$ represents the high-frequency transformation matrix for \mathbf{L}_{t1} and 2) $g(\mathbf{Z}) = \|\mathbf{V}_H^{-1}\mathbf{Z}\|_F^2$, where $\mathbf{V}_H^{-1} = (\mathbf{V}^{-1})_{N-k_c:N,:}$ represents the high-frequency transformation matrix for \mathbf{W}_{t1} . Here, \mathbf{U}_H^{-1} and \mathbf{V}_H^{-1} are the ideal high-pass filters with the cutoff eigenvalues λ_{k_c} and γ_{k_c} , respectively.

Although the penalty with an ideal high-pass filter is intuitive, it has two drawbacks: first, it requires the eigenvalue decomposition for the graph shift operator (e.g., \mathbf{L}_{t1} or \mathbf{W}_{t1}), which requires a high computational complexity for large-scale graph; second, it requires the selection of cutoff eigenvalue λ_{k_c} or γ_{k_c} , i.e., the cutoff frequency of the high-pass filter.

Based on the fact that the high-frequency component of $\tilde{\mathbf{Z}}$ is almost equal to zero, we have that the value of $\sum_{k=1}^N \sum_{m=1}^M h_m \lambda_k^m \|\tilde{\mathbf{Z}}_k\|_2^2$ is very small: first, for the small k , although $\lambda_k \neq 0$, however, $\tilde{\mathbf{Z}}_k \rightarrow \mathbf{0}$; second, for the large k , although $\tilde{\mathbf{Z}}_k \neq \mathbf{0}$, however, $\lambda_k \rightarrow 0$ with $\lambda_N = 0$. Therefore,

we can set $g(\mathbf{Z})$ as

$$\begin{aligned} g(\mathbf{Z}) &= \sum_{k=1}^N \sum_{m=1}^M h_m \lambda_k^m \|\tilde{\mathbf{Z}}_k\|_2^2 = \sum_{m=1}^M h_m \text{Tr}(\tilde{\mathbf{Z}}^T \Lambda^m \tilde{\mathbf{Z}}) \\ &= \text{Tr} \left(\sum_{m=1}^M h_m \mathbf{Z}^T \mathbf{L}_{t1}^m \mathbf{Z} \right) = \text{Tr}(\mathbf{Z}^T H(\mathbf{L}_{t1}) \mathbf{Z}) \end{aligned} \quad (30)$$

where $H(\mathbf{L}_{t1}) = \sum_{m=1}^M h_m \mathbf{L}_{t1}^m$. When $M = 1$, $g(\mathbf{Z})$ degenerates to the $h_1 \text{TV}_{\mathbf{L}_{t1}}(\mathbf{Z})$.

Second, for the sparsity regularization of $f(\Delta)$, it means that Δ only exists on a small part of vertices and remains zero on other vertices. Therefore, it requires that the number of nonzero rows in Δ , i.e., $\|\Delta\|_{2,0}$, is very small. To meet this requirement, $f(\Delta)$ can be chosen in different forms, such as the $\ell_{2,0}$ -norm [63], the $\ell_{2,1}$ -norm [64], the $\ell_{2,p}$ -norm with $p \in (0, 1)$ [65], and the $\ell_{\infty,1}$ -norm [66], [67].

Combining the spectral constraint $g(\mathbf{Z})$ and the sparsity regularization $f(\Delta)$, we have the decomposition-based SDA-HCD model as follows:

$$\min_{\mathbf{Z}, \Delta} \text{Tr}(\mathbf{Z}^T H(\mathbf{L}_{t1}) \mathbf{Z}) + \alpha f(\Delta) \quad \text{s.t. } \mathbf{Y} = \mathbf{Z} + \Delta. \quad (31)$$

From (31), it can be found that the regressed \mathbf{Z} and the original \mathbf{Y} are in the same domain for two reasons: first, \mathbf{Z} is separate from \mathbf{Y} with $\mathbf{Z} = \mathbf{Y} - \Delta$, and only a few of them are different (Δ is row sparse), i.e., $\mathbf{Z}_i = \mathbf{Y}_i$ ideally holds for most $i = 1, \dots, N$; second, for the changed \mathbf{Z}_i , the model

constrains it to be similar to its neighbors by $g(\mathbf{Z})$, which prevents anomalous \mathbf{Z}_i .

F. Optimization of SDA-HCD

By using the alternating direction method of multipliers (ADMM), the augmented Lagrangian function of (31) can be written as

$$\begin{aligned} \mathcal{L}(\mathbf{Z}, \Delta, \mathbf{R}) = & \text{Tr}(\mathbf{Z}^T H(\mathbf{L}_{t1}) \mathbf{Z}) + \text{Tr}(\mathbf{R}^T (\mathbf{Y} - \mathbf{Z} - \Delta)) \\ & + \frac{\mu}{2} \|\mathbf{Y} - \mathbf{Z} - \Delta\|_F^2 + \alpha f(\Delta) \end{aligned} \quad (32)$$

where $\mathbf{R} \in \mathbb{R}^{N \times M_y}$ is a Lagrange multiplier and $\mu > 0$ is a penalty parameter. The minimization problem of (32) can be solved by the alternating direction method, which iteratively updates one variable at a time and fixes the others.

Z-Subproblem: Given the current points $(\mathbf{Z}^t, \Delta^t, \mathbf{R}^t)$ at the t th iteration, the minimization of (32) with respect to \mathbf{Z} can be formulated as

$$\begin{aligned} \mathbf{Z}^{t+1} = \arg \min_{\mathbf{Z}} & \left\{ \text{Tr}(\mathbf{Z}^T H(\mathbf{L}_{t1}) \mathbf{Z}) - \text{Tr}((\mathbf{R}^t)^T \mathbf{Z}) \right. \\ & \left. + \frac{\mu}{2} \|\mathbf{Z} - \mathbf{Y} + \Delta^t\|_F^2 \right\}. \end{aligned} \quad (33)$$

It can be solved by taking the first-order derivative of the objective function to zero, and then, \mathbf{Z} can be updated by

$$\mathbf{Z}^{t+1} = (2H(\mathbf{L}_{t1}) + \mu \mathbf{I}_N)^{-1} (\mu \mathbf{Y} - \mu \Delta^t + \mathbf{R}^t) \quad (34)$$

where $\mathbf{I}_N \in \mathbb{R}^{N \times N}$ represents an identity matrix.

Δ-Subproblem: Give the fixed points $(\mathbf{Z}^{t+1}, \Delta^t, \mathbf{R}^t)$, the minimization of (32) with respect to Δ can be formulated as

$$\begin{aligned} \Delta^{t+1} = \arg \min_{\Delta} & \left\{ \alpha f(\Delta) - \text{Tr}((\mathbf{R}^t)^T \Delta) \right. \\ & \left. + \frac{\mu}{2} \|\Delta + \mathbf{Z}^{t+1} - \mathbf{Y}\|_F^2 \right\} \end{aligned} \quad (35)$$

which can be solved by the proximal operator as

$$\Delta^{t+1} = \text{prox}_{\frac{\alpha}{\mu f}}(\mathbf{Q}^{t+1}) \quad (36)$$

with $\mathbf{Q}^{t+1} = \mathbf{Y} - \mathbf{Z}^{t+1} + (\mathbf{R}^t/\mu)$, and the proximal operator is defined as

$$\text{prox}_{\beta f}(\mathbf{b}) = \arg \min_{\mathbf{x}} f(\mathbf{x}) + \frac{1}{2\beta} \|\mathbf{x} - \mathbf{b}\|_F^2. \quad (37)$$

Depending on different regularization forms of $f(\Delta)$, we have different closed-form solutions for updating Δ^{t+1} .

If we choose $f(\Delta) = \|\Delta\|_{2,0}$, we have

$$\Delta_i^{t+1} = \begin{cases} \mathbf{0}, & \text{if } \|\mathbf{Q}_i^{t+1}\|_2^2 \leq \frac{2\alpha}{\mu} \\ \mathbf{Q}_i^{t+1}, & \text{otherwise.} \end{cases} \quad (38)$$

If we choose the $\ell_{2,1}$ -norm of $f(\Delta) = \|\Delta\|_{2,1}$, which is a convex relaxation of $\|\Delta\|_{2,0}$, the closed-form solution of (37) can be obtained by using [68, Lemma 3.3]

$$\Delta_i^{t+1} = \max \left\{ \|\mathbf{Q}_i^{t+1}\|_2 - \frac{\alpha}{\mu} \right\} \frac{\mathbf{Q}_i^{t+1}}{\|\mathbf{Q}_i^{t+1}\|_2} \quad (39)$$

where we follow the convention $0 \cdot (0/0) = 0$.

Algorithm 2 SDA-HCD

Input: Signal \mathbf{Y} , graph G_{t1} , parameters of α, μ, ξ^0 .

Initialize: Set $\Delta^0, \mathbf{R}^0 = \mathbf{0}$, and calculate $H(\mathbf{L}_{t1})$.

Repeat:

1: Update \mathbf{Z} according to (34).

2: Update Δ according to (38), (39), or (40).

3: Update \mathbf{R} according to (41).

Until stopping criterion is met.

Output: The regressed signal \mathbf{Z} and changed signal Δ .

If we have known, in prior, the size of changed regions, i.e., the row sparsity level of Δ , we can construct a forced constraint as $f(\Delta) = \begin{cases} 0, & \text{if } \|\Delta\|_{2,0} \leq \tau \\ \infty, & \text{otherwise,} \end{cases}$ and then, Δ^{t+1} can be updated by the hard thresholding operator as

$$\Delta_i^{t+1} = \begin{cases} \mathbf{Q}_i^{t+1}, & \text{if } i \in p^\tau \\ \mathbf{0}, & \text{otherwise} \end{cases} \quad (40)$$

where p^τ is the top τ values' indices vector of $\{\|\mathbf{Q}_i^{t+1}\|_2 | i = 1, \dots, N_S\}$ with descending order.

Multiplier Updating: Finally, with the fixed points $(\mathbf{Z}^{t+1}, \Delta^{t+1}, \mathbf{R}^t)$, the Lagrangian multiplier can be updated as

$$\mathbf{R}^{t+1} = \mathbf{R}^t + \mu(\mathbf{Y} - \mathbf{Z}^{t+1} - \Delta^{t+1}). \quad (41)$$

The procedure of solving problem (31) is summarized in Algorithm 2. The algorithm terminates when the maximal number of iterations is reached or the relative difference between two iteration results $\xi^{t+1} < \xi^0$, where $\xi^{t+1} = (\|\Delta^{t+1} - \Delta^t\|_F) / (\|\Delta^t\|_F)$.

G. DI and CM Calculation

Once the regressed signal is computed from Algorithm 2, the regression image $\hat{\mathbf{Z}}$ can be obtained by extracting the pixel value in \mathbf{Z} when G_{t2} is a patchwise graph or extracting the mean features in \mathbf{Z} when G_{t2} is a superpixelwise graph. With the changed signal Δ output by Algorithm 2, we can obtain the DI as $DI_{m,n} = (\sum_{c=1}^{C_y} (\hat{y}(m, n, c) - \hat{z}(m, n, c))^2)^{1/2}$ for patchwise graph and $DI_{m,n} = \|\Delta_i\|_2, (m, n) \in \Omega_i$ for superpixelwise graph. The binary CM solution can be regarded as an image segmentation problem by using the thresholding method or the clustering method, such as the Otsu threshold method [57], K-means clustering [58], FCM clustering [59], or the MRF-based segmentation method [31].

VI. DISCUSSION

A. Connection Between VDF-HCD and Some Graph-Based Methods

In INLPG [48] and IRG-McS [41], the patchwise graphs and superpixelwise graphs are constructed for heterogeneous images, respectively, and then, the graph projection is used to compute the DIs as follows:

$$d_i^x = \frac{1}{K} \left(\sum_{j' \in \mathcal{N}_i^x} dist_{i,j'}^x - \sum_{j \in \mathcal{N}_i^x} dist_{i,j}^x \right)$$

$$d_i^y = \frac{1}{K} \left(\sum_{j' \in \mathcal{N}_i^x} \text{dist}_{i,j'}^y - \sum_{j \in \mathcal{N}_i^y} \text{dist}_{i,j}^y \right) \quad (42)$$

which measures the structure difference by how different the two KNN position sets of \mathcal{N}_i^x and \mathcal{N}_i^y are in the images. With the definition of the adjacent matrix \mathbf{A} of the KNN graph, (42) can be rewritten as

$$\begin{aligned} d_i^x &= \frac{1}{K} \sum_{j=1}^N [(\mathbf{A}_{t2} - \mathbf{A}_{t1}) \odot \text{dist}^x]_{i,j} \\ d_i^y &= \frac{1}{K} \sum_{j=1}^N [(\mathbf{A}_{t1} - \mathbf{A}_{t2}) \odot \text{dist}^y]_{i,j}. \end{aligned} \quad (43)$$

Then, we can find that (43) is a special case of (20) and (21) with $\mathbf{S} = \mathbf{W}^{\text{avg}}$ and $m = 1$, that is, INLPG and IRG-McS only consider the first-order information of the graph and ignore the exploitation of higher order information. At the same time, INLPG and IRG-McS do not take a GSP perspective on HCD issues as this article does, which leads to a limited application of these methods.

In [44], a pointwise approach based on graph theory is proposed for homogeneous CD of SAR images, which constructs a pointwise graph on a set of characteristic points and then calculates the change level of each vertex as

$$f_i = \frac{1}{\sum_{j \in \mathcal{N}_i^x} w_{i,j}^{t1}} \sum_{j \in \mathcal{N}_i^x} w_{i,j}^{t1} |\log \bar{x}_j - \log \bar{y}_j| \quad (44)$$

where \bar{x}_j and \bar{y}_j represent the mean value of a small patch around the j th vertex in images $\hat{\mathbf{X}}$ and $\hat{\mathbf{Y}}$, respectively. Equation (44) measures the difference of different signals on the same graph, which is similar to $\mathbf{f} = \mathbf{P}_{t1}(\log \mathbf{X} - \log \mathbf{Y})$. However, we can find that (44) cannot be applied to HCD because of the direct comparison between pixels of images, which will cause the leakage of heterogeneous data. At the same time, the high-order information and the influence of the changed vertices are not taken into account in this method.

As mentioned above, the proposed VDF-HCD compares the structures of images, so it is not sensitive to the interference factors in some complicated homogeneous CD problems, such as illumination, season, and noise. In addition, it has been demonstrated in INLPG [48] that the structure difference-based operator is more robust to noise than the traditional difference operator (and log-ratio operator) in the homogeneous CD of optical images (and SAR images). Therefore, the method proposed in this article can be directly extended to the homogeneous CD. At the same time, fusing the proposed method with other methods (such as the graph-based method in [44]) may further improve the detection performance.

B. Observing VDF-HCD in the Spectral Domain

In the VDF-HCD, we measure the change level similarly as $\mathbf{d}^y = H(\mathbf{S}_{t1})\mathbf{Y} - H(\mathbf{S}_{t2})\mathbf{Y}$. If we choose $H(\mathbf{S}) = \mathbf{L}$, we have $\mathbf{L}_{t1}\mathbf{Y} = \mathbf{U}\Lambda\mathbf{U}^{-1}\mathbf{Y}$. The transfer function of \mathbf{L}_{t1} is $H(\Lambda) = \Lambda$ with $h(\lambda_k) = \lambda_k$, which is a linear amplification function, as shown in the Fig. 4(c). $H(\Lambda) = \Lambda$ gives

a large value for high frequency (i.e., large eigenvalue as Definition 1), and then, it acts as a high-pass filter for the graph signal. Substituting the signal decomposition model of $\mathbf{Y} = \mathbf{Z} + \Delta$, we have $GFT(\mathbf{L}_{t1}\mathbf{Y}) = \Lambda(\tilde{\mathbf{Z}} + \tilde{\Delta})$. Then, the low-frequency components of $\tilde{\mathbf{Y}}$ are reduced by $H(\Lambda)$, and the high-frequency components of $\tilde{\mathbf{Y}}$ (i.e., $\tilde{\Delta}$) are amplified by $H(\Lambda)$. As \mathbf{Y} is a low-pass signal on G_{t2} , we have $\mathbf{L}_{t2}\mathbf{Y} \approx \mathbf{0}$. Therefore, $\mathbf{d}^y = \mathbf{L}_{t1}\mathbf{Y} - \mathbf{L}_{t2}\mathbf{Y}$ mainly contains information about Δ , which can be used to measure the changes.

Similarly, if we choose $H(\mathbf{S}) = \mathbf{W}^{\text{avg}}$, we have $H(\Gamma) = \Gamma$, which gives a large value for low frequency (i.e., large eigenvalue as Definition 2), and then, it acts as a low-pass filter for the graph signal. As \mathbf{Y} is the low-pass signal on G_{t2} , we have $\mathbf{W}_{t2}^{\text{avg}}\mathbf{Y} \approx \mathbf{Y}$. With the $GFT(\mathbf{W}_{t1}^{\text{avg}}\mathbf{Y}) = \Gamma(\tilde{\mathbf{Z}} + \tilde{\Delta})$, the high-frequency components of $\tilde{\mathbf{Y}}$ (i.e., $\tilde{\Delta}$) are reduced by $H(\Lambda)$. Then, $\mathbf{W}_{t1}^{\text{avg}}\mathbf{Y}$ can be regarded as an approximate \mathbf{Z} . Therefore, $\mathbf{d}^y = \mathbf{W}_{t1}^{\text{avg}}\mathbf{Y} - \mathbf{W}_{t2}^{\text{avg}}\mathbf{Y} \approx \mathbf{Z} - \mathbf{Y}$ can be used to measure the changes. In fact, when we choose $\mathbf{L} = \mathbf{I}_N - \mathbf{W}^{\text{avg}}$, we have $H(\Gamma) = \mathbf{I}_N - H(\Lambda)$.

C. How to Choose the Filter $H(\mathbf{S})$

For the choices of the $H(\mathbf{S})$ for vertex and spectral domain-based methods, we need to construct a low-pass filter for $H(\mathbf{W})$ in VDF-HCD and a high-pass filter for $H(\mathbf{L})$ in VDF-HCD and SDA-HCD based on the above analysis.

Suppose that $\Psi(\Lambda)$ is the desired graph transfer function of a filter; we need to use the $H(\mathbf{L}) = \sum_{m=1}^M h_m \mathbf{L}^m$ to approximate the filter. Two methods can be used for the design of this spectral domain filter [52], [55]: first, using the least-squares approximation of $\mathbf{h} = (\Theta_\lambda^T \Theta_\lambda)^{-1} \Theta_\lambda^T \psi$ with $\mathbf{h} = [h_1, \dots, h_M]^T$ being the vector of system coefficients to be estimated, Θ_λ being the Vandermonde matrix form of the eigenvalues λ_k , and $\psi = [\Psi(\lambda_1), \dots, \Psi(\lambda_N)]^T$ being the diagonal vector of $\Psi(\Lambda)$; second, using the polynomial approximation of M degree, e.g., the Chebyshev polynomial series, such as the filters used in Fig. 4(c).

D. KNN Graph Construction

We can find that the requirements for the KNN graph are that: first, G_{t1} can represent the structure of the image; second, \mathbf{X} is the low-pass signal on the graph G_{t1} . These correspond to the two challenges of the KNN graph: the choice of K and the weighting metric.

First, a very small K is not appropriate, e.g., $K = 1$, each vertex is connected with its nearest neighbor. In this case, although \mathbf{X} is definitely the low-pass signal on the graph G_{t1} , G_{t1} is not robust. In this case, each vertex can only get information from its nearest neighbor, which will cause many unnecessary disconnected subgraphs in G_{t1} . Besides, this type of graph cannot adequately characterize the structure of $\hat{\mathbf{X}}$, which also means that, perhaps, other images that differ from $\hat{\mathbf{X}}$ will also be the low-pass signal in G_{t1} . On the other hand, a very large K is also not appropriate, e.g., $K = N$, the complete graph, i.e., each vertex accepts information from all other vertices whether they are similar or not. In this case, we can easily find that \mathbf{X} is not a low-pass signal on the graph, or it has a large passband at least.

Here, we recommend the adaptive probability graph with the model

$$\begin{aligned} \min_{\mathbf{W}_{t1}} \quad & \sum_{i=1}^N \sum_{j=1}^N \text{dist}_{i,j}^{\mathbf{x}} w_{i,j}^{t1} + \mathcal{R}(\mathbf{W}_{t1}, \boldsymbol{\beta}) \\ \text{s.t. } & 0 \leq w_{i,j}^{t1} \leq 1, \quad \mathbf{W}_{t1}\mathbf{1}_N = \mathbf{1}_N \end{aligned} \quad (45)$$

where $\text{dist}_{i,j}^{\mathbf{x}} = \|\mathbf{x}_i - \mathbf{x}_j\|_2^2$, and the regularization of $\mathcal{R}(\mathbf{W}_{t1}, \boldsymbol{\beta})$ aims to make the graph smooth and avoids the trivial solution. For example, if we ignore this $\mathcal{R}(\mathbf{W}_{t1}, \boldsymbol{\beta})$, then each vertex only connects with its nearest vertex with probability 1.

We present two different regularizations for \mathbf{W} .

- 1) The weighted ℓ_2 -norm as $\mathcal{R}(\mathbf{W}_{t1}, \boldsymbol{\beta}) = \sum_{i=1}^N \sum_{j=1}^N \beta_i (w_{i,j}^{t1})^2$ used in [31], which uses the parameters β_i to determine the number of neighbors k_i of each vertex based on k -selection strategy. Therefore, we can find that G_{t1} is a data-dependent KNN graph with adaptive neighbor selection and adaptive weight calculation.
- 2) The entropy regularizer of $\mathcal{R}(\mathbf{W}_{t1}, \boldsymbol{\beta}) = \sum_{i=1}^N \sum_{j=1}^N \beta_i w_{i,j}^{t1} \log w_{i,j}^{t1}$ similar as [69], which attempts to maximize the information entropy of \mathbf{W}_{t1} and uses the parameters β_i to adjust the distribution of weights. In particular, we can find that G_{t1} constructed by (45) is very suitable for the SDA-HCD model (31): the regularization term $\sum_{i=1}^N \sum_{j=1}^N \text{dist}_{i,j}^{\mathbf{x}} w_{i,j}^{t1} = 2\text{Tr}(\mathbf{X}^T \mathbf{L}_{t1} \mathbf{X})$ in the objective function of graph construction model (45) is consistent with the penalty term $\text{Tr}(\mathbf{Z}^T H(\mathbf{L}_{t1}) \mathbf{Z})$ in the regression model.

E. Extended Graphs for SDA-HCD

In this article, we construct the KNN graph for each image to capture the structure information. An important reason why the KNN graph was chosen is that it can distinguish between the changed and unchanged signals in the spectral domain: the original \mathbf{X} and regressed \mathbf{Z} are low-pass signals in the KNN graph G_{t1} , while the high-frequency components of $\tilde{\mathbf{Y}}$ are introduced by the changed signal Δ . Similarly, if we can find such a graph G' that the changed and unchanged signals can be discriminated in the spectral domain with G' , then more regularization terms $g(\mathbf{Z})$ can be added to the regression model (29).

If we construct a K -farthest neighbor (KFN) graph $G'_{t1} = \{\mathcal{V}'_{t1}, \mathcal{E}'_{t1}, \mathbf{W}'_{t1}\}$ for the preevent image, where each vertex is connected with its KFN, i.e., $(i, j) \in \mathcal{E}'_{t1}$ if and only if $\text{dist}_{i,j}^{\mathbf{x}}$ is among the K -largest elements of the vector $\text{dist}_i^{\mathbf{x}}$ or the vector $\text{dist}_j^{\mathbf{x}}$, in this way, the change rate of signal \mathbf{X} between the vertices on the edges of G'_{t1} is very rapid, that is, the TV of $\text{TV}_{\mathbf{L}'_{t1}}(\mathbf{X})$ is very large. Therefore, the high-frequency component of $\tilde{\mathbf{X}}$ in the graph G'_{t1} is also large, so as to the regression signal \mathbf{Z} , as shown in Fig. 7. To exploit this property, we can add a KFN graph-induced repulsive regularization term $g(\mathbf{Z}) = -\text{TV}_{\mathbf{L}'_{t1}}(\mathbf{Z})$ or $g(\mathbf{Z}) = \sum_{(i,j) \in \mathcal{E}'_{t1}} w_{i,j}^{t1} \exp(-\text{dist}_{i,j}^{\mathbf{z}})$ in the regression model (29), which requires neighboring nodes of \mathbf{Z} connected by the KFN graph G'_{t1} to be further apart

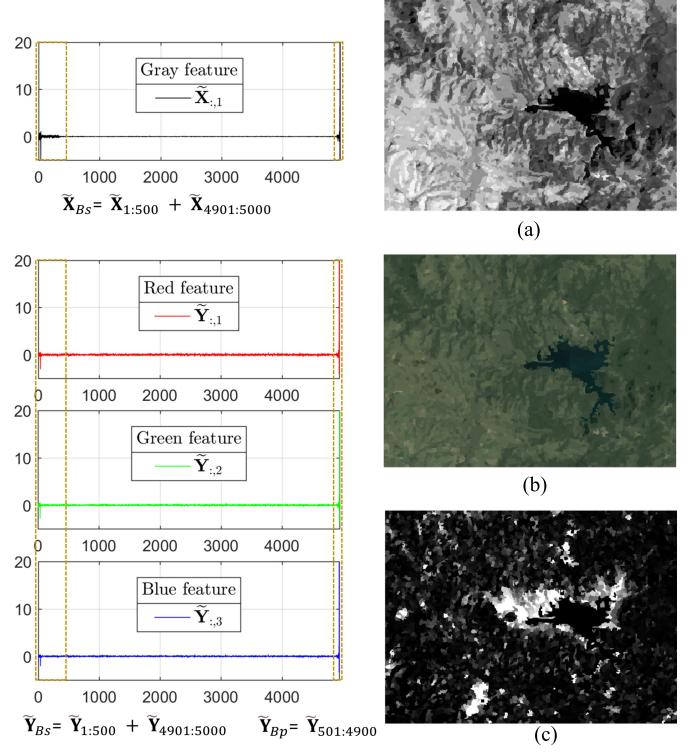


Fig. 7. (b) Regression image and (c) DI generated by the band-stop and bandpass filters defined on the KFN graph G'_{t1} , respectively, where \mathbf{X} , \mathbf{Y} , and G'_{t1} are constructed from the preevent and postevent images in Fig. 6. The high-frequency component of $\tilde{\mathbf{X}}$ in the graph G'_{t1} , i.e., $\tilde{\mathbf{X}}_{4901:5000}$, is large. The approximate regressed image can be obtained with partial low- and high-frequency components, i.e., $\tilde{\mathbf{Y}}_{BS}$, and the DI can be calculated with partial mid-frequency components, i.e., $\tilde{\mathbf{Y}}_{Bp}$. (a) $\tilde{\mathbf{X}}'$ with $\mathbf{X}' = \mathbf{U}\tilde{\mathbf{X}}_{BS}$. (b) $\tilde{\mathbf{Y}}'$ with $\mathbf{Y}' = \mathbf{U}\tilde{\mathbf{Y}}_{BS}$. (c) $\tilde{\mathbf{D}}'\mathbf{I}'$ with $\mathbf{Z}' = \mathbf{U}\tilde{\mathbf{Y}}_{Bp}$.

(because they do not represent the same type of object). Fig. 7 shows an example of the usage of KFN in HCD. Similarly, other types of graph properties can be extended in the proposed GSP-based HCD framework.

F. Frameworks of GSP-Based CD

Here, we give the general frameworks for the CD problem (with both homogeneous and heterogeneous multitemporal remote sensing images) based on the GSP with vertex domain filtering and spectral domain analysis.

1) *Framework of VDF-Based CD: Step 1:* Construct the KNN graph and graph signals.

Step 2: Choose the filter $H(\mathbf{S})$, and calculate the DI.

Step 3: Segment the DI to obtain the changed and unchanged indices.

Step 4: Repeat steps 1–3 to reduce the change influence until the algorithm converges.

In step 1, it requires that the constructed graph can represent the structure of the image and the graph signal can represent the characteristics of the area denoted by each vertex. In step 2, $H(\mathbf{S})$ should be able to reflect the differences in signals on different graphs, i.e., be able to reflect the structural properties of the graphs, e.g., low-pass filter for $H(\mathbf{W}^{\text{avg}})$ and $H(\mathbf{P})$, and a high-pass filter for $H(\mathbf{L}^{\text{rw}})$ according to

TABLE I
DESCRIPTION OF THE SEVEN HETEROGENEOUS DATASETS

Dataset	Sensor	Size (pixels)	Date	Location	Event (& Spatial resolution)
#1	Landsat-5/Google Earth	300 × 412 × 1(3)	Sept. 1995 - July 1996	Sardinia, Italy	Lake expansion (30m.)
#2	Landsat-5/Landsat-8	1534 × 808 × 7(10)	Aug. 2011 - June 2013	Texas, USA	Forest fire (30m.)
#3	Pleiades/WorldView2	2000 × 2000 × 3(3)	May 2012 - July 2013	Toulouse, France	Construction (0.52m.)
#4	Spot/NDVI	990 × 554 × 3(1)	1999 - 2000	Gloucester, England	Flooding (\approx 25m.)
#5	Radsats-2/Google Earth	593 × 921 × 1(3)	June 2008 - Sept. 2012	Shuguang Village, China	Building construction (8m.)
#6	Landsat-8/Sentinel-1A	875 × 500 × 11(3)	Jan. 2017 - Feb. 2017	Sutter County, USA	Flooding (\approx 15m.)
#7	Radsats-2/Google Earth	343 × 291 × 1(3)	June 2008 - Sept. 2010	Yellow River, China	Embankment change (8m.)

the spectral analysis. Meanwhile, in the homogeneous CD, $\mathbf{d}^x = H(\mathbf{S}_{t1})\mathbf{X} - H(\mathbf{S}_{t1})\mathbf{Y}$ and $\mathbf{d}^y = H(\mathbf{S}_{t2})\mathbf{X} - H(\mathbf{S}_{t2})\mathbf{Y}$ can also be fused in the DI calculation. In step 3, many existing methods can be selected for this binary segmentation problem. Finally, the change influence is reduced by an iterative strategy that cuts off the propagation from changed vertices to their neighboring vertices.

2) *Framework of SDA-Based CD: Step 1:* Construct the graph and graph signals.

Step 2: Choose the penalties of $g(\mathbf{Z})$ and $f(\Delta)$ to obtain the regression model of (29).

Step 3: Solve the minimization problem to obtain the changed signal.

Step 4: Segment the DI to obtain the final CM.

In step 1, it requires that the graph signals are distinguishable in the spectral domain of the graph. In step 2, $g(\mathbf{Z})$ is corresponding to the spectral property of the signal, such as the low-pass property of the KNN graph; $f(\Delta)$ can also be other prior knowledge-based penalties in addition to the sparsity penalty, such as the smoothness [46], low-rank, and some known labels of the semisupervised (or supervised) CD problem. In step 4, the binary classification problem can be solved by many existing methods [57], [58], [59].

VII. EXPERIMENTS

In this section, we analyze the performance of the proposed VDF-HCD and SDA-HCD. In order to assess the robustness, we use seven pairs of heterogeneous images of varying HCD conditions.

A. Experimental Setting

1) *Heterogeneous Datasets:* First, we introduce the heterogeneous datasets used in this article, as listed in Table I. These datasets contain different types of heterogeneity: multisensor image pairs (e.g., #1, #2, #3, and #4) and multisource image pairs (e.g., #5, #6, and #7), which also show quite different HCD conditions: different resolutions (from 0.52 to 30 m), different sizes (from 300 to 2000 pixels in length or width), and different change events (flooding, fire, and construction). These heterogeneous datasets are able to evaluate the generalizability and robustness of the proposed method.

2) *Metrics:* We evaluate the detection ability of the methods quantitatively through two types of metrics. First, we assess the DI by the receiver operating characteristic (ROC) and precision-recall (PR) curves, along with the areas under the

ROC curve (AUR) and PR curve (AUP) scores. Second, we assess the CM by the overall accuracy (OA), F1-measure (Fm), and Kappa coefficient (Kc), which are calculated by $OA = (TP + TN)/(TP + TN + FP + FN)$, $Fm = (2TP)/(2TP + FP + FN)$, and $Kc = (OA - PRE)/(1 - PRE)$ with

$$PRE = \frac{(TP + FN)(TP + FP) + (TN + FP)(TN + FN)}{(TP + TN + FP + FN)^2} \quad (46)$$

where TP, FP, TN, and FN represent the true positive, false positive, true negative, and false negative, respectively. Combining the comprehensive metrics (OA, Kc, and Fm) with the individual metrics (TP, FP, TN, and FN marked in different colors in the qualitative results) allows us to better assess the CM.

3) *Implementation Detail:* We use the superpixelwise KNN graph for VDF-HCD and SDA-CHD with $N = 5000$ and $K = \sqrt{N}$, use the mean, median, and variance values of superpixel to construct the graph signals of \mathbf{X} and \mathbf{Y} for simplicity, and choose the adaptive model (45) with weighted ℓ_2 -norm to construct the graphs. We choose the graph shift operator as $\mathbf{S} = \mathbf{W}^{\text{avg}}$ and set $H(\mathbf{W}^{\text{avg}})$ to obtain an approximate low-pass filter for VDF-HCD, whose transfer function in the spectral domain is a fourth-order ($M = 4$) polynomial approximation of the truncation function $H(\Gamma) = ((\text{sign}(\Gamma - \gamma_{cf}) + 1)/2)$ with the cutoff frequency being $\gamma_{cf} = 0.9$. We choose $\ell_{2,1}$ -norm of $f(\Delta) = \|\Delta\|_{2,1}$, set the balancing parameter $\alpha = 0.05$ for the regression model (31), and choose $H(\mathbf{L}_{t1}) = \mathbf{L}_{t1} + h_2 \mathbf{L}_{t1}^2 + h_3 \mathbf{L}_{t1}^3$ with $h_2 = h_3 = 1$ for SDA-HCD. We leave the discussion about $H(\mathbf{W}^{\text{avg}})$ and $H(\mathbf{L}_{t1})$ in Section VI-C.

B. Experimental Results

1) *Image Regression of SDA-HCD:* In the first experiment, we verify the regression performance of the SDA-HCD by comparing it with four image regression baseline HCD methods, including HPT [15] (40% of the unchanged pixels are used as training samples in our experiment), AMD-IR [23], PSGM [19], and SCASC [31]. Fig. 8 shows the regression images generated by different methods on all the heterogeneous datasets, which transforms the preevent image to the domain of the postevent image. Intuitively, these methods basically complete the image translation. That is, in the unchanged area, the regression image and the postevent image are similar, while they appear very different in the changed area, which can be used to detect the changes. Visually, we can see that SDA-HCD performs better on Datasets #1, #3, and #7; PSGM performs better on Dataset #5 and worse on Dataset #7; and

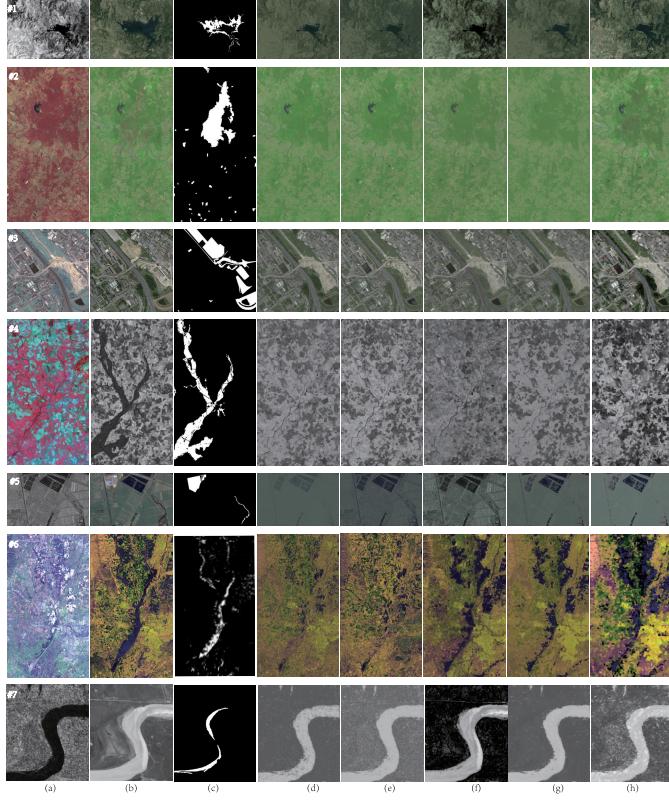


Fig. 8. Regression images generated by different methods on all the heterogeneous datasets. From (top) to (bottom), they correspond to Datasets #1–#7, respectively. From (left) to (right) are (a) preevent image, (b) postevent image, (c) ground truth, (d) regression image of HPT, (e) regression image of AMD-IR, (f) regression image of PSGM, (g) regression image of SCASC, and (h) regression image of SDA-HCD.

TABLE II
AUR AND AUP OF DIS GENERATED BY DIFFERENT METHODS ON THE HETEROGENEOUS DATASETS. THE HIGHEST SCORES ARE HIGHLIGHTED IN BOLD

Measures	Methods	Datasets							Average
		#1	#2	#3	#4	#5	#6	#7	
AUR	HPT [15]	0.889	0.920	0.770	0.947	0.911	0.928	0.964	0.904
	AMD-IR [23]	0.795	0.918	0.708	0.927	0.922	0.867	0.859	0.857
	PSGM [19]	0.837	0.898	0.777	0.933	0.975	0.911	0.728	0.866
	SCASC [31]	0.885	0.932	0.793	0.936	0.968	0.887	0.969	0.910
	VDF-HCD (F ^x)	0.859	0.946	0.814	0.903	0.896	0.903	0.960	0.897
	VDF-HCD (F ^y)	0.900	0.937	0.805	0.951	0.971	0.897	0.923	0.912
AUP	HPT [15]	0.373	0.575	0.355	0.806	0.470	0.506	0.602	0.527
	AMD-IR [23]	0.155	0.536	0.264	0.741	0.564	0.289	0.216	0.395
	PSGM [19]	0.593	0.526	0.405	0.745	0.794	0.484	0.152	0.528
	SCASC [31]	0.383	0.566	0.458	0.636	0.695	0.447	0.597	0.540
	VDF-HCD (F ^x)	0.571	0.850	0.468	0.659	0.479	0.463	0.653	0.592
	VDF-HCD (F ^y)	0.615	0.800	0.521	0.768	0.810	0.284	0.609	0.630
	SDA-HCD	0.531	0.603	0.571	0.648	0.785	0.413	0.688	0.606

HPT performs better on Datasets #4 and #6. Also, we can find that the superpixel-based regression method has a block smoothing effect, such as the regression images of SCASC and SDA-HCD on Datasets #4 and #6.

2) *Difference Images*: In order to evaluate the ability of the algorithms (VDF-HCD and SDA-HCD) to measure change, we show the DI generated by different methods in Fig. 9 and plot the corresponding ROC and PR curves in Fig. 10. The AUR of ROC curves and the AUP of PR curves are listed in Table II.

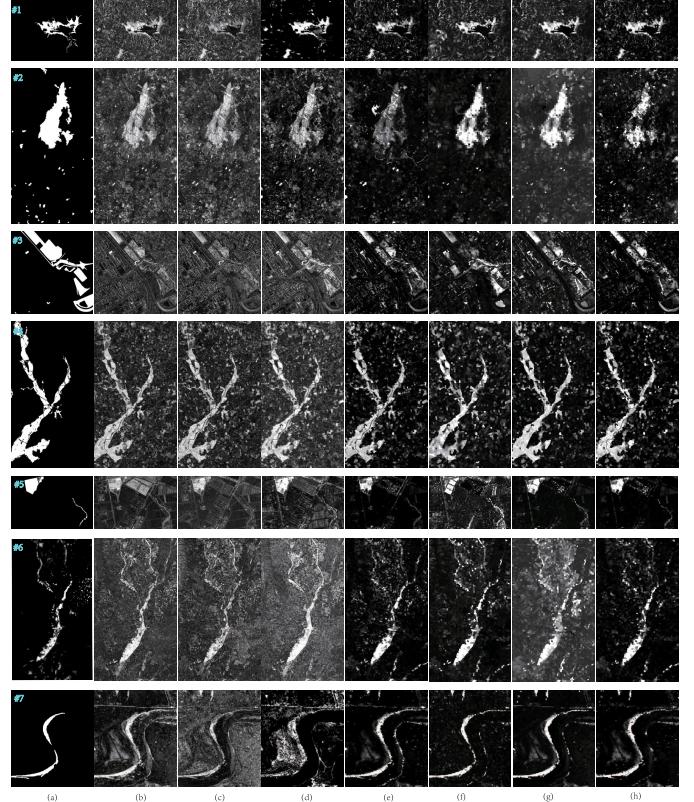


Fig. 9. DI of different methods on heterogeneous datasets. From (top) to (bottom), they correspond to Datasets #1 to #7, respectively. From (left) to (right) are (a) ground truth, (b) DI of HPT, (c) DI of AMD-IR, (d) DI of PSGM, (e) DI of SCASC, (f) forward DI of VDF-HCD, (g) backward DI of VDF-HCD, and (h) DI of SDA-HCD.

In the DI of Datasets #1 and #5, the backward DI of VDF-HCD (\mathbf{f}^y) and the DI of PSGM perform better than others, as shown in Figs. 9(b)–(h) and 10. In Dataset #2, the forward and backward DI of VDF-HCD gain higher scores of AUR and AUP. In the DI of Datasets #3 and #7, SDA-HCD performs best, and VDF-HCD comes second, as shown by the ROC and PR curves in Fig. 10. In the DI of Datasets #4 and #6, HPT performs better than other methods. However, these results are attributed to the fact that HPT employs a large amount of labeled unchanged samples (40%). Meanwhile, we can also find that the DI generated by PSGM, SCASC, and SDA-HCD in Fig. 9 are sparse (e.g., especially in the DI of Datasets #1, #2, and #7), which is due to the prior sparsity-based regularization used in these methods.

From these results, we can see that the VDF-HCD and SDA-HCD can obtain high-quality DI in datasets of #1, #2, #4, #5, and #7, which gains a larger AUR and AUP scores, as reported in Table II. Therefore, it is able to obtain a satisfactory CM by directly segmenting the DI with a simple thresholding method (e.g., Otsu thresholding [57]) or clustering method (e.g., K-means clustering [58]) for Datasets #1, #2, #4, #5, and #7. For the Datasets #3 and #6, they contain more types of ground objects than other datasets (e.g., buildings, grass, roads, pitches, farmland, rivers, forests, roads, and mountains), these objects are very unevenly proportioned, and the resolution of the images in Dataset #3 is very high (0.52 m), as shown in

Fig. 8. It means that images of Datasets #3 and #6 contain many categories of features, which poses difficulties in capturing image structure and completing structure comparison.

By comparing the forward and backward DIs of VDF-HCD on each dataset, it can be observed that they provide complementary information, which, in turn, can be fused to obtain better detection results. This complementary advantage of DIs can also be seen visually in Fig. 9, where a more obvious contrast of changed vertex and unchanged vertex can be obtained by directly summing the \mathbf{f}^x and \mathbf{f}^y . Comparing SCASC and SDA-HCD, we can find that they produce similar DIs because they both belong to the structured graph-based regression methods; however, there are differences in the performance of their DIs in Fig. 9 and Table II, i.e., SDA-HCD outperforms SCASC, which is due to the use of high-order information and spectral analysis in SDA-HCD.

3) *Change Maps*: Third, to test the ability of detecting changes, another five recently proposed SOTA methods for HCD are adopted for additional comparison, including M3CD [22], FPMS [32], CICM [11], NPSG [14], and IRG-McS [41]. We use the default parameters in their codes, which are also consistent with the related papers.

The visual results of the binary CMs generated by different methods on all the heterogeneous datasets are shown in Fig. 11. Intuitively, the results generated by these comparison methods can generally reflect the main information of changes. To be specific, the CMs provided by HPT and AMD-IR are affected by the salt-and-pepper noise, as shown in Fig. 11(a) and (b), especially on Dataset #3, which is because their pixel-to-pixel regression process is not robust enough to cope with complex and variable HCD conditions. As a result, a large number of unchanged pixels are misclassified into changed ones, e.g., the CMs of HPT on Datasets #3 and #5 and the CMs of AMD-IR on Datasets #1, #3, and #7. The PSGM achieves relatively good performances on some datasets, such as Datasets #1 and #5, but also has higher FP on some datasets, such as Dataset #7. The M3CD, FPMS, CICM, and NPSG do not perform robustly enough, and their performance degrades dramatically on some complex HCD scenes, such as M3CD, FPMS, CICM, and NPSG on Dataset #2, M3CD on Dataset #6, and CICM on Dataset #7. In the results of Datasets #6 and #7, many false alarms appear in the CM of M3CD, while, in the results of Datasets #2 and #3, many missed detection appear in the CM of CICM. SCASC and IRG-McS can achieve satisfactory CMs with less FP on Datasets #2, #4, and #7, thanks to the exploitation of graph-based structure consistency between heterogeneous images. Nevertheless, due to the fact that SCASC and IRG-McS ignore the higher order neighborhood information hidden in the graphs, making their performance not as good as VDF-HCD and SDA-HCD. On the whole, the proposed GSP-based methods can suppress false alarms, reduce missed detection simultaneously, and outperform other comparison methods. These performances can be attributed to three main factors: 1) the graph-based method is more robust to different HCD conditions, such as scenes, noises, and sensors; 2) the high-order neighborhood information used in the methods helps to capture the structure of complex regions accurately and effectively; and 3) the negative influences of

changes are reduced in the propose methods, e.g., cutting off the signal transmission from the changed vertices by VDF-HCD and directly separating changed signal with the signal decomposition model by SDA-HCD.

Table III reports the quantitative evaluation results of comparison methods on the datasets. It is clear that the quantitative results are consistent with the visual analysis of Fig. 11. We can see that the proposed methods obtain better results than other SOTA methods on most datasets. For example, the VDF-HCD gains the highest Kc and Fm in Datasets #1, #2, #5, and #7, and the SDA-HCD gains the highest scores in Dataset #3 and obtains very competitive performance on other datasets. The average scores (OA, Kc, and Fm) of VDF-HCD and SDA-HCD on all evaluated datasets are about (0.956, 0.687, 0.710) and (0.956, 0.663, 0.686), respectively, which are higher than other comparison methods. This demonstrates that the proposed methods can efficiently improve HCD performance.

Finally, in order to further compare the performance of the proposed GSP-based HCD methods, i.e., the VDF-HCD and SDA-HCD, the results obtained by some representative and SOTA methods [21], [27], [37], [38], [39], [70], [71], [72], [73], [74], [75], [76], [77], [78], [79], [80], [81] are summarized in Table IV, except for M3CD [22], FPMS [32], CICM [11], NPSG [14], IRG-McS [41], HPT [15], AMD-IR [23], PSGM [19], and SCASC [31], which have been compared in detail in this article. Among these comparison approaches, *DFR-MT* [70], *CACFL* [27], *AFL-DSR*, *DCCAE* [75], *DCCA* [76], *DPFL* [79], *LT-FT* [80], *SSL* [21], *ACE-Net* [33], *X-Net* [33], *SCCN* [39], and *ASDNN* [81] 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 IV. Because the datasets used in each paper are not identical, Table IV is not aligned. As can be seen in Table IV, the proposed GSP-based methods (VDF-HCD and SDA-HCD) consistently yield better or very competitive accuracy across different datasets by comparing with these SOTA approaches, which again demonstrates the effectiveness of the proposed GSP perspective for the HCD problem.

4) *Comparison of VDF-HCD and SDA-HCD*: From the above experiments, we can find that VDF-HCD is superior to SDA-HCD in terms of detection accuracy in most datasets, such as the average AUR and AUP in Table II, and the average OA, Kc, and Fm in Table III. However, SDA-HCD has three advantages: 1) compared to the VDF-HCD that only provides the change results, i.e., DI or CM, the regression-based SDA-HCD also provides a translated image of another moment, which has a stronger visualization effect than VDF-HCD; 2) SDA-HCD is more efficient than VDF-HCD and requires less computational time, as reported in Table VI; and 3) the model of SDA-HCD is very flexible and easy to extend as described in Section VI, which has great potential for applications.

C. Discussions

1) *Choices of $H(\mathbf{S})$* : As analyzed in Section VI-C, we need to construct a low-pass filter for $H(\mathbf{W}^{\text{avg}})$ in VDF-HCD. We use the polynomials of $H(\mathbf{W}^{\text{avg}})$ to approximate the

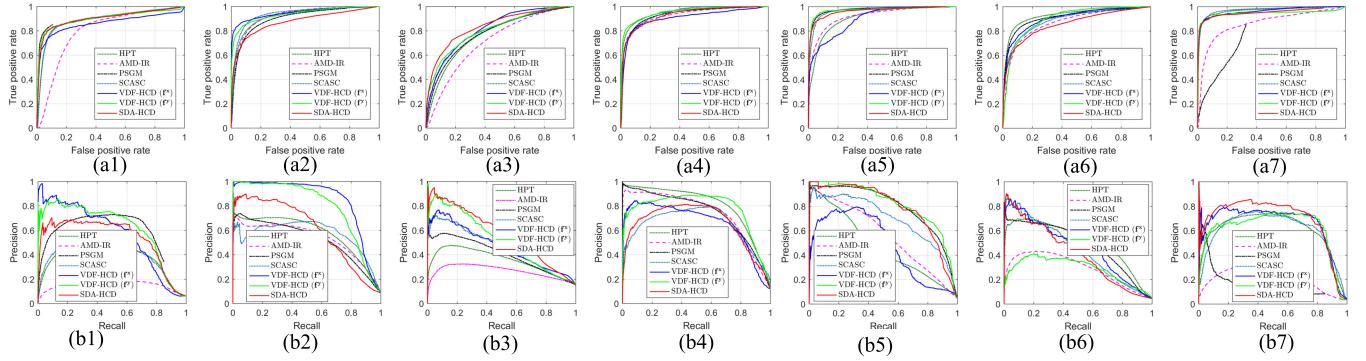


Fig. 10. ROC and PR curves of DI generated by different methods on all the heterogeneous datasets. Top row: ROC curves. Bottom row: PR curves. From (left) to (right) are the results on Datasets #1–#7, respectively. (a1) ROC curves on Dataset #1. (a2) ROC curves on Dataset #2. (a3) ROC curves on Dataset #3. (a4) ROC curves on Dataset #4. (a5) ROC curves on Dataset #5. (a6) ROC curves on Dataset #6. (a7) ROC curves on Dataset #7. (b1) PR curves on Dataset #1. (b2) PR curves on Dataset #2. (b3) PR curves on Dataset #3. (b4) PR curves on Dataset #4. (b5) PR curves on Dataset #5. (b6) PR curves on Dataset #6. (b7) PR curves on Dataset #7.

TABLE III

QUANTITATIVE MEASURES OF BINARY CMs ON THE HETEROGENEOUS DATASETS. THE HIGHEST SCORES ARE HIGHLIGHTED IN BOLD

Methods	Dataset #1			Dataset #2			Dataset #3			Dataset #4			Dataset #5			Dataset #6			Dataset #7			Average		
	OA	Kc	Fm																					
HPT [15]	0.912	0.504	0.546	0.916	0.589	0.633	0.815	0.415	0.523	0.918	0.671	0.717	0.859	0.318	0.368	0.932	0.488	0.518	0.927	0.447	0.476	0.897	0.490	0.540
AMD-IR [23]	0.799	0.255	0.328	0.901	0.547	0.598	0.724	0.259	0.411	0.898	0.607	0.664	0.950	0.572	0.597	0.822	0.236	0.291	0.686	0.116	0.17	0.826	0.370	0.437
PSGM [19]	0.961	0.682	0.703	0.928	0.631	0.670	0.857	0.473	0.558	0.922	0.675	0.719	0.977	0.744	0.756	0.908	0.383	0.422	0.832	0.124	0.173	0.912	0.530	0.572
SCASC [31]	0.947	0.593	0.621	0.961	0.751	0.772	0.892	0.464	0.516	0.949	0.771	0.800	0.979	0.741	0.751	0.961	0.479	0.500	0.977	0.700	0.711	0.952	0.643	0.667
M3CD [22]	0.963	0.669	0.689	0.896	0.009	0.044	0.863	0.405	0.481	0.915	0.588	0.636	0.962	0.602	0.622	0.575	0.021	0.077	0.856	0.158	0.204	0.861	0.350	0.393
FPMS [32]	0.925	0.552	0.588	0.893	0.001	0.012	0.838	0.215	0.296	0.962	0.816	0.837	0.938	0.569	0.597	0.947	0.329	0.356	0.979	0.544	0.553	0.926	0.432	0.463
CICM [11]	0.943	0.451	0.481	0.907	0.006	0.021	0.867	0.270	0.321	0.884	0.507	0.573	0.974	0.745	0.759	0.899	0.081	0.131	0.789	0.024	0.080	0.895	0.298	0.338
NPSG [14]	0.947	0.559	0.587	0.902	0.458	0.511	0.830	0.346	0.446	0.902	0.608	0.663	0.975	0.729	0.742	0.941	0.419	0.449	0.985	0.733	0.741	0.926	0.550	0.591
IRG-McS [41]	0.971	0.739	0.754	0.933	0.473	0.506	0.882	0.420	0.478	0.939	0.714	0.749	0.983	0.794	0.804	0.959	0.490	0.512	0.976	0.690	0.702	0.949	0.617	0.644
VDF-HCD	0.971	0.756	0.771	0.973	0.813	0.828	0.886	0.448	0.504	0.944	0.754	0.786	0.985	0.808	0.816	0.952	0.487	0.512	0.982	0.745	0.754	0.956	0.687	0.710
SDA-HCD	0.958	0.654	0.676	0.966	0.768	0.787	0.902	0.523	0.572	0.947	0.768	0.798	0.982	0.766	0.775	0.954	0.433	0.457	0.981	0.728	0.738	0.955	0.663	0.686

TABLE IV

ACCURACY RATE OF CMs GENERATED BY DIFFERENT METHODS ON DIFFERENT DATASETS. THE RESULTS OF THESE COMPARISON METHODS ARE REPORTED BY THEIR ORIGINAL PUBLISHED PAPERS, EXCEPT THAT THE RESULTS INDICATED WITH † ARE REPORTED BY YANG ET AL. [76] (THEY ARE CONSISTENT WITH THEIR OPEN SOURCE CODES IN [37]). ITALICIZED AND UNDERLINED MARKS ARE USED FOR DEEP LEARNING-BASED METHODS

Dataset #1	OA	Dataset #2	OA	Dataset #3	OA	Dataset #4	OA	Dataset #5	OA	Dataset #6	OA	Dataset #7	OA
<i>DFR-MT</i> [70]	0.975	<i>VDF-HCD</i> 0.973	<i>SDA-HCD</i> 0.966	<i>VDF-HCD</i> 0.886	<i>SDA-HCD</i> 0.947	<i>VDF-HCD</i> 0.944	<i>SDA-HCD</i> 0.947						
<i>CACFL</i> [27]	0.975	<i>VDF-HCD</i> 0.973	<i>SDA-HCD</i> 0.966	<i>VDF-HCD</i> 0.886	<i>SDA-HCD</i> 0.947	<i>VDF-HCD</i> 0.944	<i>SDA-HCD</i> 0.947						
<i>ALSC</i> [71]	0.965	<i>DCCAE</i> [75]	0.957	<i>VDF-HCD</i> 0.944	<i>SDA-HCD</i> 0.947								
<i>SDA-HCD</i> 0.958	<i>DCCA</i> [76]	0.947	<i>VDF-HCD</i> 0.944	<i>SDA-HCD</i> 0.947	<i>VDF-HCD</i> 0.944								
<i>MDS</i> [72]	0.942	<i>VDF-HCD</i> 0.944	<i>SDA-HCD</i> 0.947										
<i>AFL-DSR</i> [73]	0.929	<i>VDF-HCD</i> 0.944	<i>SDA-HCD</i> 0.947										
<i>RMN</i> [74]	0.847	<i>VDF-HCD</i> 0.944	<i>SDA-HCD</i> 0.947										
<i>DPFL</i> [79]	0.987	<i>VDF-HCD</i> 0.985	<i>SDA-HCD</i> 0.982										
<i>GIR-MRF</i> [77]	0.959	<i>VDF-HCD</i> 0.985	<i>SDA-HCD</i> 0.982										
<i>AFL-DSR</i> [73]	0.980	<i>VDF-HCD</i> 0.985	<i>SDA-HCD</i> 0.982										
<i>CACFL</i> [27]	0.979	<i>VDF-HCD</i> 0.985	<i>SDA-HCD</i> 0.982										
<i>ALSC</i> [71]	0.944	<i>VDF-HCD</i> 0.985	<i>SDA-HCD</i> 0.982										
<i>ACE-Net</i> [33]	0.915	<i>VDF-HCD</i> 0.985	<i>SDA-HCD</i> 0.982										
<i>X-Net</i> [33]	0.911	<i>VDF-HCD</i> 0.985	<i>SDA-HCD</i> 0.982										
<i>SSL</i> [21]	0.902	<i>VDF-HCD</i> 0.985	<i>SDA-HCD</i> 0.982										
<i>RMN</i> [74]	0.884	<i>VDF-HCD</i> 0.985	<i>SDA-HCD</i> 0.982										

low-pass filter with the transfer function of $H(\Gamma) = \text{sign}(\Gamma - \gamma_{cf}) + 1/2$. It can be proven that the eigenvalues of \mathbf{W}^{avg} satisfy $\gamma_k \in [-1, 1]$, $k = 1, 2, \dots, N$, and the larger eigenvalue represents the lower frequency according to Definition 2. In Fig. 12, we plot the different polynomial transfer functions to approximate the functions of $H(\Gamma)$ with different cutoff frequencies γ_{cf} . In these filters, two parameters are important: the polynomial order M that corresponds to the hop number of neighbors that each vertex can reach and the cutoff frequency that determines the bandwidth of the filter.

In Fig. 13, we verify the polynomial order M (from 1 to 5) and cutoff frequency γ_{cf} (from 0.5 to 1 with an interval of 0.1) in the proposed VDF-HCD. Two remarks can be observed in Fig. 13.

- 1) The higher order M usually brings better results than the $M = 1$ (the average Fm value is only 0.658 when $M = 1$), which demonstrates the effectiveness of the high-order information. However, a larger M is also not recommended for two reasons: first, creating redundancy; second, computational complex.

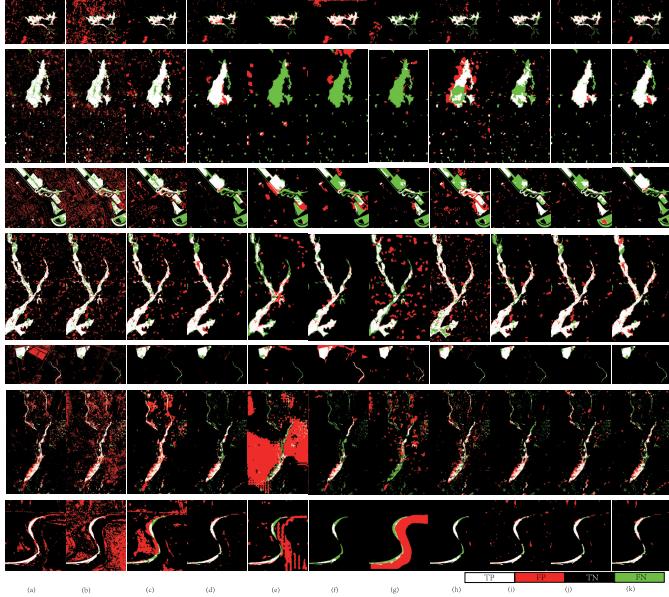


Fig. 11. Binary CMs of different methods on heterogeneous datasets. From (top) to (bottom), they correspond to Datasets #1 to #7, respectively. From (left) to (right) are CMs generated by (a) HPT, (b) AMD-IR, (c) PSGM, (d) SCASC, (e) M3CD, (f) FPMS, (g) CICM, (h) NPSG, (i) IRG-MCs, (j) VDF-HCD, and (k) SDA-HCD. In the binary CM, white: true positives (TP); red: FPs; black: TNs; and green: FNs.

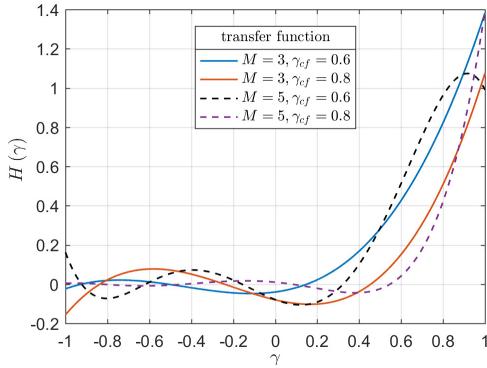


Fig. 12. Polynomial transfer function to approximate the $H(\Gamma)$ with different polynomial orders M and different cutoff frequencies γ_{cf} .

- 2) γ_{cf} controls the bandwidth of the low-pass filter, as shown in Fig. 12. Although it can be estimated accurately by the GFT, this process requires eigenvalue decomposition, which is very complex for a large-scale graph. However, as can be seen from Fig. 13 that the VDF-HCD is robust for parameter γ_{cf} , there is a wide range of acceptable values for γ_{cf} . For example, in our experiments, we fixed $\gamma_{cf} = 0.9$ for simplicity.

In the SDA-HCD, the regression model requires the spectral constraint $g(\mathbf{Z})$ (30) to penalize high-frequency components of $\tilde{\mathbf{Z}}_H$ such that the regressed \mathbf{Z} have the same spectral property (low-pass) as \mathbf{X} in the KNN graph G_{t1} , i.e., $\tilde{\mathbf{Z}}_H \approx \mathbf{0}$. If we decompose $H(\mathbf{L}_{t1}) = H^{1/2}(\mathbf{L}_{t1})H^{1/2}(\mathbf{L}_{t1})$, then we have $g(\mathbf{Z}) = \|H^{1/2}(\mathbf{L}_{t1})\mathbf{Z}\|_F^2$, where $H^{1/2}(\mathbf{L}_{t1})$ represents a graph filter with the transfer function: $H^{1/2}(\mathbf{A}) = (\sum_{m=1}^M h_m \Lambda^m)^{1/2}$. Therefore, it requires that $H^{1/2}(\mathbf{L}_{t1})$ is an

TABLE V
QUANTITATIVE RESULTS OF SDA-HCD WITH DIFFERENT $H(\mathbf{L}_{t1})$ 'S

SDA-HCD with $H(\mathbf{L}_{t1})$	Average scores				
	AUR	AUP	OA	Kc	Fm
\mathbf{L}_{t1}	0.896	0.542	0.946	0.634	0.655
$\mathbf{L}_{t1} + \mathbf{L}_{t1}^2$	0.898	0.591	0.953	0.655	0.678
$\mathbf{L}_{t1} + \mathbf{L}_{t1}^2 + \mathbf{L}_{t1}^3$	0.900	0.606	0.955	0.663	0.686
$\mathbf{L}_{t1} + \mathbf{L}_{t1}^2 + \mathbf{L}_{t1}^3 + \mathbf{L}_{t1}^4$	0.898	0.594	0.955	0.657	0.680

approximate high-pass filter. In Fig. 14, we plot different functions of $H^{1/2}(\lambda) = (\sum_{m=1}^M h_m \lambda^m)^{1/2}$. In these functions, we can find that the higher order polynomial transfer function $H^{1/2}(\lambda) = (\lambda + \lambda^2 + \lambda^3)^{1/2}$ is closer to a high-pass function than $H^{1/2}(\lambda) = \sqrt{\lambda}$. This also intuitively explains why SDA-HCD, which uses higher order information and spectral analysis, and performs better than the SCASC, as illustrated by the results of Figs. 8–11 and Tables II and III.

In Table V, we list the quantitative results (average scores on all the evaluated datasets) of SDA-HCD with different $H(\mathbf{L}_{t1})$, including \mathbf{L}_{t1} , $\mathbf{L}_{t1} + \mathbf{L}_{t1}^2$, $\mathbf{L}_{t1} + \mathbf{L}_{t1}^2 + \mathbf{L}_{t1}^3$, and $\mathbf{L}_{t1} + \mathbf{L}_{t1}^2 + \mathbf{L}_{t1}^3 + \mathbf{L}_{t1}^4$. With respect to the first-order $H(\mathbf{L}_{t1}) = \mathbf{L}_{t1}$, high-order $H(\mathbf{L}_{t1})$ improves by 6.4% and 3.1% on average AUP and Fm in Table V, respectively. By contrast, the fourth-order $H(\mathbf{L}_{t1})$ deteriorates the performance a little bit on the average scores, which is perhaps caused by the excessively narrow bandwidth of the transfer function.

2) *Choices of K*: In the KNN graph, the number of nearest neighbors plays an important role. In Fig. 15, we vary K from 30 to 130 with step 20. It can be found that the detection performance is not very sensitive to the value of K . Of course, a very small K is not appropriate, which will make the graph not robust and, thus, affect the change measurements. For example, for the unchanged vertex i , when one of its neighbor $j' \in \mathcal{N}_i^Y$ is polluted by noise or changed, $\mathbf{X}_{j'}$ will be different from \mathbf{X}_i , and it brings errors in \mathbf{d}_i^X of (7). In this case, a larger K will reduce the influence of $\mathbf{X}_{j'}$ by average weighting. This can also be illustrated in Figs. 1(a) and 3(a), where the unchanged fourth and seventh vertices with three neighbors in G_{t1} and G_{t2} are less affected by changed fifth vertex than the others, e.g., the first and second vertices in G_{t1} , and the sixth and ninth vertices in G_{t2} , with two neighbors. On the other hand, a particularly large K is also not appropriate, which will make the change measurements less discriminative. For example, when K exceeds the actual number of the real similar neighbors of the vertex, then this vertex will receive information from the vertices that are not related to it. In the extreme case of $K = N$, we have the measurement of (14) equals to 0 for all vertices. Therefore, we empirically set $K = \sqrt{N}$, which comes from the work of KNN-based density estimation [82], [83], [84] and KNN classification [85].

3) *Parameter of α*: The balancing parameter of α is used to control the sparsity level of Δ in Algorithm 2 according to the update of Δ with (38), (39), or (40), which should be selected according to the proportion of the changed area. Generally, the smaller the changed regions, the larger α should be. We evaluate the sensitivity of the SDA-HCD model to α

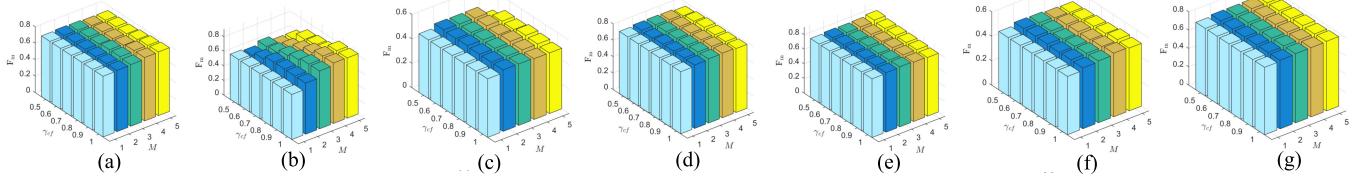


Fig. 13. Sensitivity analysis of $H(\Gamma)$ with different M 's and γ_{cf} 's in VDF-HCD on all heterogeneous datasets. (a) Dataset #1. (b) Dataset #2. (c) Dataset #3. (d) Dataset #4. (e) Dataset #5. (f) Dataset #6. (g) Dataset #7.

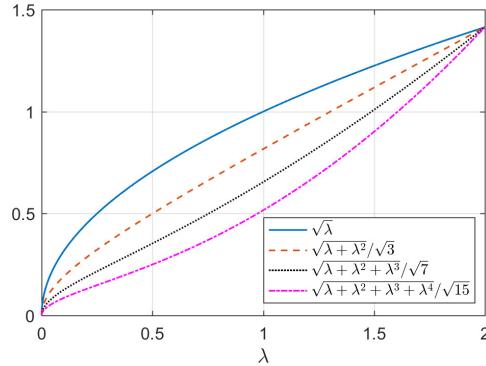


Fig. 14. Different functions of $H^{(1/2)}(\lambda) = (\sum_{m=1}^M h_m \lambda^m)^{(1/2)}$.

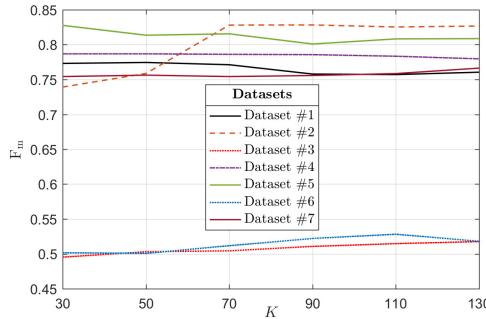


Fig. 15. Sensitivity analysis of parameter K in VDF-HCD.

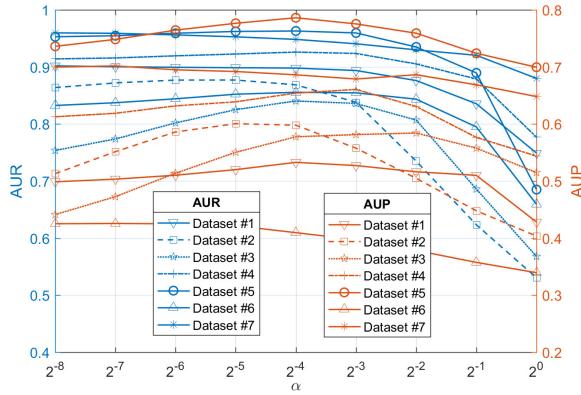


Fig. 16. Sensitivity analysis of parameter α in SDA-HCD.

in Fig. 16, which is assessed by measuring the AUR and AUP of DI with different α (from 2^{-8} to 1 with the ratio of 2). Based on Fig. 16, we fix $\alpha = 0.05$ (i.e., $\alpha \approx 2^{-4.3}$) in our experiments for simplicity.

TABLE VI
COMPUTATIONAL TIME (SECONDS) OF VDF-HCD AND SDA-HCD

Data sets	N	t_{VDF}	t_{SDA}
Dataset #7 $343 \times 291 \times 1(3)$	5000	26.49	6.84
	10000	100.13	27.20
Dataset #2 $2000 \times 2000 \times 3(3)$	5000	29.91	10.22
	10000	104.33	29.77

4) *Computational Analysis:* The main computational complexity of the proposed methods is concentrating on graph construction and structure comparison (VDF-HCD) or image regression (SDA-HCD). For the former, calculating the distance matrix between all the patches or superpixels requires $\mathcal{O}((M_x + M_y)N^2/2)$, sorting the distance matrix by column to construct the graph requires $\mathcal{O}(N^2 \log N)$, and calculating the $H(\mathbf{S})$ requires $\mathcal{O}((M - 1)N^3)$. For the structure comparison of VDF-HCD in Algorithm 1, the MRF cosegmentation requires $\mathcal{O}(2N_R N^2)$ for each iteration with the worst case [41], where N_R is the number of edges in the R -adjacency neighbor system of the MRF cosegmentation model. For the image regression model of SDA-HCD in Algorithm 2, updating \mathbf{Z} with (34) requires $\mathcal{O}(N^3)$, updating Δ requires $\mathcal{O}(M_y N)$ by using the closed-form proximal operator (37), and updating \mathbf{R} with (41) requires $\mathcal{O}(M_y N)$. Although the complexity of Algorithm 2 is very high, which requires $\mathcal{O}(N^3)$ for each iteration, two acceleration strategies are available to improve the efficiency of Algorithm 2 as introduced in [31]: computing the matrix inversion of $(2H(\mathbf{L}_{t1}) + \mu\mathbf{I}_N)^{-1}$ offline in advance or solving the \mathbf{Z} -subproblem (34) with the preconditioned conjugate gradient (PCG) method.

Table VI reports the computational time of VDF-HCD and SDA-HCD with different N 's on Datasets #7 and #2, which is performed in MATLAB 2016a running on a Windows desktop with Intel Core i7-8700K CPU. As can be seen in Table VI, the running time is mainly determined by the graph scale N , and SDA-HCD is more efficient than VDF-HCD because it does not require iterations to eliminate the effects of changes as in VDF-HCD.

VIII. CONCLUSION

Motivated by the performance gained by the graph-based methods, we investigate the inner workings of HCD methods and propose a new strategy for solving the HCD problem from the perspective of GSP. By defining the graphs and graph signals, the changes between heterogeneous images manifest

themselves in two aspects: the structure difference between graphs and the signal difference on the graph. Thereby, we can compare the responses of the two signals on different filters to detect the changes.

Therefore, we propose two frameworks for the HCD problem with GSP, i.e., VDF-HCD from the vertex domain and SDA-HCD from the spectral domain. First, VDF-HCD measures the structure difference between graphs by comparing the output signals of the same input signal on filters defined on different graphs. VDF-HCD can explore the high-order information hidden in the graphs by using different filters and alleviate the negative influence of changes by using an iterative strategy that cuts off the signal propagation from changed vertices to their neighboring vertices. Second, we analyze the GSP for HCD from the spectral domain. We first show the spectral properties of the heterogeneous images on the same graph and illustrate that it is the changes between images that cause the differences in their spectral properties. Based on the graph spectral analysis, we propose a signal decomposition-based SDA-HCD, which decomposes the source signal into the regressed signal and changed signal, and constrains the spectral property of the regressed signal.

We hope that the proposed method can inspire research on multimodal remote sensing images from the perspective of GSP, such as image fusion and image registration. This is particularly important for how to combine and rejuvenate traditional methods, especially as deep learning-based methods are currently being used systematically. In addition, we only consider the simple graph, linear filtering, and GFT in this article. Our future work includes studying hypergraphs, graph wavelet analysis, and graph neural networks on the HCD problem.

ACKNOWLEDGMENT

The authors would like to thank the researchers for their friendly sharing of heterogeneous change detection codes and datasets. They would also like to thank the editors and anonymous reviewers for their constructive suggestions that improved the presentation of this work.

REFERENCES

- [1] A. Singh, “Review article digital change detection techniques using remotely-sensed data,” *Int. J. Remote Sens.*, vol. 10, no. 6, pp. 989–1003, 1989.
- [2] S. Liu, D. Marinelli, L. Bruzzone, and F. Bovolo, “A review of change detection in multitemporal hyperspectral images: Current techniques, applications, and challenges,” *IEEE Geosci. Remote Sens. Mag.*, vol. 7, no. 2, pp. 140–158, Jun. 2019.
- [3] Z. Lv, T. Liu, J. A. Benediktsson, and N. Falco, “Land cover change detection techniques: Very-high-resolution optical images: A review,” *IEEE Geosci. Remote Sens. Mag.*, vol. 10, no. 1, pp. 44–63, Mar. 2022.
- [4] D. Wen et al., “Change detection from very-high-spatial-resolution optical remote sensing images: Methods, applications, and future directions,” *IEEE Geosci. Remote Sens. Mag.*, vol. 9, no. 4, pp. 68–101, Mar. 2021.
- [5] H. Dong, W. Ma, L. Jiao, F. Liu, and L. Li, “A multiscale self-attention deep clustering for change detection in SAR images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–16, 2022.
- [6] Z. Lv, T. Liu, C. Shi, and J. A. Benediktsson, “Local histogram-based analysis for detecting land cover change using VHR remote sensing images,” *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 7, pp. 1284–1287, Jul. 2021.
- [7] J. Qu, S. Hou, W. Dong, Y. Li, and W. Xie, “A multilevel encoder-decoder attention network for change detection in hyperspectral images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–13, 2022.
- [8] Z. Lv, G. Li, Z. Jin, J. A. Benediktsson, and G. M. Foody, “Iterative training sample expansion to increase and balance the accuracy of land classification from VHR imagery,” *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 139–150, Jan. 2021.
- [9] Z. Lv, F. Wang, G. Cui, J. A. Benediktsson, T. Lei, and W. Sun, “Spatial-spectral attention network guided with change magnitude image for land cover change detection using remote sensing images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–12, 2022.
- [10] S. N. Anfinsen, L. T. Luppino, M. A. Hansen, G. Moser, and S. B. Serpico, “Unsupervised heterogeneous change detection in radar images by cross-domain affinity matching,” in *Proc. IEEE Radar Conf.*, Sep. 2020, pp. 1–6.
- [11] R. Touati, “Détection de changement en imagerie satellitaire multimodale,” Ph.D. dissertation, Département d’Informatique et de Recherche Opérationnelle, Université de Montréal, Montreal, QC, Canada, 2019.
- [12] F. Bovolo, S. Marchesi, and L. Bruzzone, “A framework for automatic and unsupervised detection of multiple changes in multitemporal images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 6, pp. 2196–2212, Jun. 2012.
- [13] G. Moser and S. B. Serpico, “Generalized minimum-error thresholding for unsupervised change detection from SAR amplitude imagery,” *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 10, pp. 2972–2982, Oct. 2006.
- [14] Y. Sun, L. Lei, X. Li, H. Sun, and G. Kuang, “Nonlocal patch similarity based heterogeneous remote sensing change detection,” *Pattern Recognit.*, vol. 109, Jan. 2021, Art. no. 107598.
- [15] Z. Liu, G. Li, G. Mercier, Y. He, and Q. Pan, “Change detection in heterogeneous remote sensing images via homogeneous pixel transformation,” *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 1822–1834, Apr. 2018.
- [16] J. R. Jensen, E. W. Ramsey, H. E. Mackey, Jr., E. J. Christensen, and R. R. Sharitz, “Inland wetland change detection using aircraft MSS data,” *Photogram. Eng. Remote Sens.*, vol. 53, no. 5, pp. 521–529, 1987.
- [17] X. Jiang, G. Li, X.-P. Zhang, and Y. He, “A semisupervised Siamese network for efficient change detection in heterogeneous remote sensing images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–18, 2022.
- [18] J. Wu et al., “A multiscale graph convolutional network for change detection in homogeneous and heterogeneous remote sensing images,” *Int. J. Appl. Earth Observ. Geoinf.*, vol. 105, Jan. 2021, Art. no. 102615.
- [19] Y. Sun, L. Lei, X. Li, X. Tan, and G. Kuang, “Patch similarity graph matrix-based unsupervised remote sensing change detection with homogeneous and heterogeneous sensors,” *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 6, pp. 4841–4861, Jun. 2021.
- [20] S. Saha, P. Ebel, and X. X. Zhu, “Self-supervised multisensor change detection,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–10, 2022.
- [21] Y. Chen and L. Bruzzone, “Self-supervised change detection in multiview remote sensing images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–12, 2022.
- [22] R. Touati, M. Mignotte, and M. Dahmane, “Multimodal change detection in remote sensing images using an unsupervised pixel pairwise-based Markov random field model,” *IEEE Trans. Image Process.*, vol. 29, pp. 757–767, 2020.
- [23] L. T. Luppino, F. M. Bianchi, G. Moser, and S. N. Anfinsen, “Unsupervised image regression for heterogeneous change detection,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 9960–9975, Dec. 2019.
- [24] J. Prendes, M. Chabert, F. Pascal, A. Giros, and J. Y. Tourneret, “A new multivariate statistical model for change detection in images acquired by homogeneous and heterogeneous sensors,” *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 799–812, Mar. 2015.
- [25] X. Jiang, G. Li, Y. Liu, X.-P. Zhang, and Y. He, “Change detection in heterogeneous optical and SAR remote sensing images via deep homogeneous feature fusion,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 1551–1566, 2020.
- [26] J. Liu, W. Zhang, F. Liu, and L. Xiao, “A probabilistic model based on bipartite convolutional neural network for unsupervised change detection,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–14, 2022.
- [27] Y. Wu, J. Li, Y. Yuan, A. K. Qin, Q.-G. Miao, and M.-G. Gong, “Commonality autoencoder: Learning common features for change detection from heterogeneous images,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 9, pp. 4257–4270, Sep. 2021.
- [28] L. Wan, Y. Xiang, and H. You, “An object-based hierarchical compound classification method for change detection in heterogeneous optical and SAR images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 9941–9959, Dec. 2019.

- [29] L. Wan, Y. Xiang, and H. You, "A post-classification comparison method for SAR and optical images change detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 7, pp. 1026–1030, Jul. 2019.
- [30] Y. Wu, Z. Bai, Q. Miao, W. Ma, Y. Yang, and M. Gong, "A classified adversarial network for multi-spectral remote sensing image change detection," *Remote Sens.*, vol. 12, no. 13, p. 2098, Jun. 2020.
- [31] Y. Sun, L. Lei, D. Guan, M. Li, and G. Kuang, "Sparse-constrained adaptive structure consistency-based unsupervised image regression for heterogeneous remote-sensing change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–14, 2022.
- [32] M. Mignotte, "A fractal projection and Markovian segmentation-based approach for multimodal change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 11, pp. 8046–8058, Nov. 2020.
- [33] L. T. Luppino et al., "Deep image translation with an affinity-based change prior for unsupervised multimodal change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–22, 2021.
- [34] Z.-G. Liu, Z.-W. Zhang, Q. Pan, and L.-B. Ning, "Unsupervised change detection from heterogeneous data based on image translation," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–13, 2022.
- [35] X. Li, Z. Du, Y. Huang, and Z. Tan, "A deep translation (GAN) based change detection network for optical and SAR remote sensing images," *ISPRS J. Photogramm. Remote Sens.*, vol. 179, pp. 14–34, Sep. 2021.
- [36] G. Mercier, G. Moser, and S. B. Serpico, "Conditional copulas for change detection in heterogeneous remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 5, pp. 1428–1441, May 2008.
- [37] M. Volpi, G. Camps-Valls, and D. Tuia, "Spectral alignment of multi-temporal cross-sensor images with automated kernel canonical correlation analysis," *J. Photogram. Remote Sens.*, vol. 107, pp. 50–63, Sep. 2015.
- [38] R. Touati and M. Mignotte, "An energy-based model encoding nonlocal pairwise pixel interactions for multisensor change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 1046–1058, Oct. 2017.
- [39] J. Liu, M. Gong, K. Qin, and P. Zhang, "A deep convolutional coupling network for change detection based on heterogeneous optical and radar images," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 3, pp. 545–559, Mar. 2016.
- [40] H. Li, M. Gong, M. Zhang, and Y. Wu, "Spatially self-paced convolutional networks for change detection in heterogeneous images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 4966–4979, 2021.
- [41] Y. Sun, L. Lei, D. Guan, and G. Kuang, "Iterative robust graph for unsupervised change detection of heterogeneous remote sensing images," *IEEE Trans. Image Process.*, vol. 30, pp. 6277–6291, 2021.
- [42] A. Sanfeliu, R. Alquézar, J. Andrade, J. Climent, F. Serratosa, and J. Vergés, "Graph-based representations and techniques for image processing and image analysis," *Pattern Recognit.*, vol. 35, no. 3, pp. 639–650, 2002.
- [43] M. T. Pham, G. Mercier, and J. Michel, "Pointwise graph-based local texture characterization for very high resolution multispectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 5, pp. 1962–1973, May 2015.
- [44] M.-T. Pham, G. Mercier, and J. Michel, "Change detection between SAR images using a pointwise approach and graph theory," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 4, pp. 2020–2032, Apr. 2016.
- [45] J. Wang, X. Yang, X. Yang, L. Jia, and S. Fang, "Unsupervised change detection between SAR images based on hypergraphs," *ISPRS J. Photogramm. Remote Sens.*, vol. 164, pp. 61–72, Jun. 2020.
- [46] D. A. Jimenez-Sierra, D. A. Quintero-Olaya, J. C. Alvear-Munoz, H. D. Benitez-Restrepo, J. F. Florez-Ospina, and J. Chanussot, "Graph learning based on signal smoothness representation for homogeneous and heterogeneous change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–16, 2022.
- [47] D. A. Jimenez-Sierra, H. D. Benitez-Restrepo, H. D. Vargas-Cardona, and J. Chanussot, "Graph-based data fusion applied to: Change detection and biomass estimation in Rice crops," *Remote Sens.*, vol. 12, no. 17, p. 2683, Aug. 2020.
- [48] Y. Sun, L. Lei, X. Li, X. Tan, and G. Kuang, "Structure consistency-based graph for unsupervised change detection with homogeneous and heterogeneous remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–21, 2022.
- [49] A. Gavili and X.-P. Zhang, "On the shift operator, graph frequency, and optimal filtering in graph signal processing," *IEEE Trans. Signal Process.*, vol. 65, no. 23, pp. 6303–6318, Dec. 2017.
- [50] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3042–3054, Jun. 2014.
- [51] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, Apr. 2013.
- [52] L. Stankovic, D. P. Mandic, M. Dakovic, I. Kisil, E. Sejdic, and A. G. Constantinides, "Understanding the basis of graph signal processing via an intuitive example-driven approach," *IEEE Signal Process. Mag.*, vol. 36, no. 6, pp. 133–145, Nov. 2019.
- [53] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, May 2018.
- [54] L. Stanković, M. Daković, and E. Sejdić, "Introduction to graph signal processing," in *Vertex-Frequency Analysis of Graph Signals*. Cham, Switzerland: Springer, 2019, pp. 3–108.
- [55] L. Stanković et al., "Data analytics on graphs Part II: Signals on graphs," *Found. Trends® Mach. Learn.*, vol. 13, nos. 2–3, pp. 1–177, 2020.
- [56] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.
- [57] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. SMC-9, no. 1, pp. 62–66, Feb. 1979.
- [58] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A K-means clustering algorithm," *Appl. Statist.*, vol. 28, no. 1, pp. 100–108, 1979.
- [59] J. C. Bezdek, W. Full, and R. Ehrlich, "FCM: The fuzzy c-means clustering algorithm," *Comput. Geosci.*, vol. 10, nos. 2–3, pp. 191–203, 1984.
- [60] P. Milanfar, "Symmetrizing smoothing filters," *SIAM J. Imag. Sci.*, vol. 6, no. 1, pp. 263–284, 2013.
- [61] S. H. Chan, T. Zickler, and Y. M. Lu, "Understanding symmetric smoothing filters: A Gaussian mixture model perspective," *IEEE Trans. Image Process.*, vol. 26, no. 11, pp. 5107–5121, Nov. 2017.
- [62] R. Sinkhorn and P. Knopp, "Concerning nonnegative matrices and doubly stochastic matrices," *Pacific J. Math.*, vol. 21, no. 2, pp. 343–348, 1967.
- [63] X. Du, F. Nie, W. Wang, Y. Yang, and X. Zhou, "Exploiting combination effect for unsupervised feature selection by $\ell_{2,0}$ norm," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 1, pp. 201–214, Jan. 2019.
- [64] J. Lu, S. Yi, J. Zhao, Y. Liang, and W. Liu, "Interpretable robust feature selection via joint $\ell_{2,1}$ -norms minimization," in *Proc. 13th Int. Conf. Mach. Learn. Comput.*, Feb. 2021, pp. 1813–1821.
- [65] Q. Wang, Q. Gao, X. Gao, and F. Nie, " $\ell_{2,p}$ -norm based PCA for image recognition," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1336–1346, Mar. 2018.
- [66] Y. Chen and A. O. Hero, "Recursive $\ell_{1,\infty}$ group lasso," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 3978–3987, Aug. 2012.
- [67] D. Jin, J. Chen, C. Richard, and J. Chen, "Online proximal learning over jointly sparse multitask networks with $\ell_{\infty,1}$ regularization," *IEEE Trans. Signal Process.*, vol. 68, pp. 6319–6335, 2020.
- [68] J. Yang, W. Yin, Y. Zhang, and Y. Wang, "A fast algorithm for edge-preserving variational multichannel image restoration," *SIAM J. Imag. Sci.*, vol. 2, no. 2, pp. 569–592, 2009.
- [69] R. Zhang, X. Li, H. Zhang, and F. Nie, "Deep fuzzy K-means with adaptive loss and entropy regularization," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 11, pp. 2814–2824, Nov. 2020.
- [70] P. Zhang, M. Gong, L. Su, J. Liu, and Z. Li, "Change detection based on deep feature representation and mapping transformation for multi-spatial-resolution remote sensing images," *Photogram. Remote Sens.*, vol. 116, pp. 24–41, Sep. 2016.
- [71] L. Lei, Y. Sun, and G. Kuang, "Adaptive local structure consistency-based heterogeneous remote sensing change detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.
- [72] R. Touati, M. Mignotte, and M. Dahmane, "Change detection in heterogeneous remote sensing images based on an imaging modality-invariant MDS representation," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 3998–4002.
- [73] R. Touati, M. Mignotte, and M. Dahmane, "Anomaly feature learning for unsupervised change detection in heterogeneous images: A deep sparse residual model," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 588–600, 2020.
- [74] R. Touati, M. Mignotte, and M. Dahmane, "A reliable mixed-norm-based multiresolution change detector in heterogeneous remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 9, pp. 3588–3601, Sep. 2019.

- [75] Y. Zhou, H. Liu, D. Li, H. Cao, J. Yang, and Z. Li, "Cross-sensor image change detection based on deep canonically correlated autoencoders," in *Proc. Int. Conf. Artif. Intell. Commun. Netw.* Cham, Switzerland: Springer, 2019, pp. 251–257.
- [76] J. Yang, Y. Zhou, Y. Cao, and L. Feng, "Heterogeneous image change detection using deep canonical correlation analysis," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 2917–2922.
- [77] Y. Sun, L. Lei, X. Tan, D. Guan, J. Wu, and G. Kuang, "Structured graph based image regression for unsupervised multimodal change detection," *ISPRS J. Photogramm. Remote Sens.*, vol. 185, pp. 16–31, Mar. 2022.
- [78] Z.-G. Liu, G. Mercier, J. Dezert, and Q. Pan, "Change detection in heterogeneous remote sensing images based on multidimensional evidential reasoning," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 1, pp. 168–172, Jan. 2014.
- [79] M. Yang, L. Jiao, F. Liu, B. Hou, S. Yang, and M. Jian, "DPFL-Nets: Deep pyramid feature learning networks for multiscale change detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 11, pp. 6402–6416, Nov. 2022.
- [80] T. Zhan, M. Gong, X. Jiang, and S. Li, "Log-based transformation feature learning for change detection in heterogeneous images," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 9, pp. 1352–1356, Sep. 2018.
- [81] W. Zhao, Z. Wang, M. Gong, and J. Liu, "Discriminative feature learning for unsupervised change detection in heterogeneous images based on a coupled neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 12, pp. 7066–7080, Dec. 2017.
- [82] D. O. Loftsgaarden and C. P. Quesenberry, "A nonparametric estimate of a multivariate density function," *Ann. Math. Statist.*, vol. 36, no. 3, pp. 1049–1051, 1965.
- [83] K. Fukunaga and L. Hostetler, "Optimization of k -nearest-neighbor density estimates," *IEEE Trans. Inf. Theory*, vol. IT-19, no. 3, pp. 320–326, May 1973.
- [84] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, vol. 31. New York, NY, USA: Springer, 2013.
- [85] P. Mitra, C. A. Murthy, and S. K. Pal, "Unsupervised feature selection using feature similarity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 3, pp. 301–312, Mar. 2002.



Yuli Sun received the B.S. degree from Xiangtan University, Xiangtan, China, in 2009, and the M.S. degree from the University of Science and Technology of China, Hefei, China, in 2014. He is currently pursuing the Ph.D. degree in information and communication engineering with the College of Electronic Science, National University of Defense Technology, Changsha, China. He has published papers in IEEE TGRS, TIP, TNNLS, ISPRS P&RS, *Pattern Recognition*, and so on. His research interests cover remote sensing image processing and multitemporal image analysis.



Lin Lei received the Ph.D. degree in information and communication engineering from the National University of Defense Technology, Changsha, China, in 2008.

She is currently a Professor with the School of Electronic Science, National University of Defense Technology. Her research interests include computer vision, remote sensing image interpretation, and data fusion.



Dongdong Guan received the B.S. degree in surveying and mapping engineering from Wuhan University, Wuhan, China, in 2013, and the M.S. degree in photogrammetry and remote sensing and the Ph.D. degree in information and communication engineering from the National University of Defense Technology, Changsha, China, in 2015 and 2019, respectively.

He currently works at the High-Tech Institute of Xi'an, Xi'an, China. His research interests include synthetic aperture radar image processing, machine learning, and computation vision in remote sensing applications.



Gangyao Kuang (Senior Member, IEEE) received the B.S. and M.S. degrees in geophysics from the Central South University of Technology, Changsha, China, in 1988 and 1991, and the Ph.D. degree in communication and information from the National University of Defense Technology, Changsha, China, in 1995.

He is currently a Professor with the School of Electronic Science, National University of Defense Technology. His research interests include remote sensing, synthetic aperture radar (SAR) image processing, change detection, SAR ground moving target indication, and classification with polarimetric SAR images.



Li Liu (Senior Member, IEEE) received the Ph.D. degree in information and communication engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2012.

From 2015 to 2016, she spent ten months visiting the Multimedia Laboratory, The Chinese University of Hong Kong, Hong Kong. From December 2016 to November 2018, she worked as a Senior Researcher at the Machine Vision Group, University of Oulu, Oulu, Finland. She is currently a Full Professor with the College of System Engineering, NUDT. Her papers have currently over 7800 citations in Google Scholar. Her research interests include computer vision, pattern recognition, and machine learning.

Dr. Liu was the Co-Chair of nine international workshops at CVPR, ICCV, and ECCV. She serves as the Area Chair of ICME'20, ICME'21, ICME'22, ACCV'20, ACCV'22, and so on. She has served as a Leading Guest Editor for special issues in *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE* (*IEEE TPAMI*) and *International Journal of Computer Vision*. She is serving as a Leading Guest Editor for *IEEE TPAMI* special issue on Learning With Fewer Labels in Computer Vision. She serves as Associate Editor for *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY*, *Pattern Recognition*, and *Pattern Recognition Letters*.