Graph Moving Object Segmentation

Jhony H. Giraldo, Sajid Javed, and Thierry Bouwmans

Abstract—Moving Object Segmentation (MOS) is a fundamental task in computer vision. Due to undesirable variations in the background scene, MOS becomes very challenging for static and moving camera sequences. Several deep learning methods have been proposed for MOS with impressive performance. However, these methods show performance degradation in the presence of unseen videos; and usually, deep learning models require large amounts of data to avoid overfitting. Recently, graph learning has attracted significant attention in many computer vision applications since they provide tools to exploit the geometrical structure of data. In this work, concepts of graph signal processing are introduced for MOS. First, we propose a new algorithm that is composed of segmentation, background initialization, graph construction, unseen sampling, and a semi-supervised learning method inspired by the theory of recovery of graph signals. Second, theoretical developments are introduced, showing one bound for the sample complexity in semi-supervised learning, and two bounds for the condition number of the Sobolev norm. Our algorithm has the advantage of requiring less labeled data than deep learning methods while having competitive results on both static and moving camera videos. Our algorithm is also adapted for Video Object Segmentation (VOS) tasks and is evaluated on six publicly available datasets outperforming several state-of-the-art methods in challenging conditions.

Index Terms—Moving object segmentation, graph signal processing, semi-supervised learning, unseen videos, video object segmentation

1 Introduction

Moving object segmentation (MOS) is a relevant topic in computer vision and video analysis. It is a pre-processing task in many applications such as intelligent visual surveillance systems of human activities in public spaces, intelligent transportation, and robotics system, among others [1]. The main objective of MOS is to separate the moving objects called foreground, from the static component called background [2], [3], [4], [5]. In the literature, MOS has been considered as a classification problem where each pixel is predicted for either background or moving object in a sequence taken from a static or moving camera, and therefore this problem is also known as moving object detection and background subtraction [6]. Many scientific efforts have been reported in the literature to improve the classical methods progressively in applications where challenges are becoming more complex. However, none of the methods can simultaneously address all the key challenges that are present in videos during long sequences as in the real cases [7]. In fact, several studies are focused on designing methods for specific challenges in MOS such as turbulence [8], dynamic backgrounds [9], and camouflaged moving objects [10]. However, many studies are limited to deal with shadows and the sequences taken from Pan-Tilt-Zoom (PTZ) cameras because of their challenging nature [1], [11], [12], [13].

• *Jhony H. Giraldo and Thierry Bouwmans are with the laboratoire MIA, Mathématiques, Image et Applications, La Rochelle Université, 17000 La Rochelle, France. E-mail: {jgiral01, tbouwman}@univ-lr.fr.*
• *Sajid Javed is with the Center for Autonomous Robotics System, Khalifa University, Abu Dhabi 127788, UAE. E-mail: sajid.javed@ku.ac.ae.*

Manuscript received 14 Feb. 2020; revised 12 Oct. 2020; accepted 28 Nov. 2020. Date of publication 9 Dec. 2020; date of current version 1 Apr. 2022. (Corresponding authors: Jhony H. Giraldo.) Recommended for acceptance by W. Liu. Digital Object Identifier no. 10.1109/TPAMI.2020.3042093

MOS methods can broadly be categorized into unsupervised and supervised learning schemes. Many unsupervised learning methods have been proposed in the literature, and they rely on background models to predict the foreground objects [5], [11], [27]. However, these methods show performance degradation in the presence of dynamic background scenes. Supervised learning methods are based on end-to-end Deep Convolutional Neural Networks (DCNNs) [12]. DCNNs-based MOS methods have demonstrated better performance than the unsupervised methods, however, the majority of these models fail to get optimal performance when employed on unseen videos (poor generalization). As an example, the FgSegNet method uses 200 images from the test video for training and the remaining frames from the same video for evaluation [28]. The performance of FgSegNet dramatically decreases when applied to unseen videos [19]. Fig. 1 shows some visual results of the State-Of-The-Art (SOTA) methods for MOS under challenging background scenarios. Despite significant efforts and competitive performance in particular challenges, there are still several open issues for the MOS task [1], [12]. 1) None of the methods can effectively handle all MOS challenges in the presence of static and moving camera sequences. 2) Some DCNNs-based MOS methods do not have a good performance on unseen videos, or their generalizations to other videos are hardly predictable [19]. 3) Deep learning methods lack theoretical guarantees and explanations about the sample complexity required during the end-to-end learning process. The classical fundamental theorem of statistical learning, involving the Vapnik–Chervonenkis dimension, bounds the sample complexity in machine learning [29]. However, this bound does not guarantee the performance in deep learning regimen because of the huge amount of parameters in common deep neural networks.

In recent years, a growing number of graph-based representation methods have been proposed for many computer

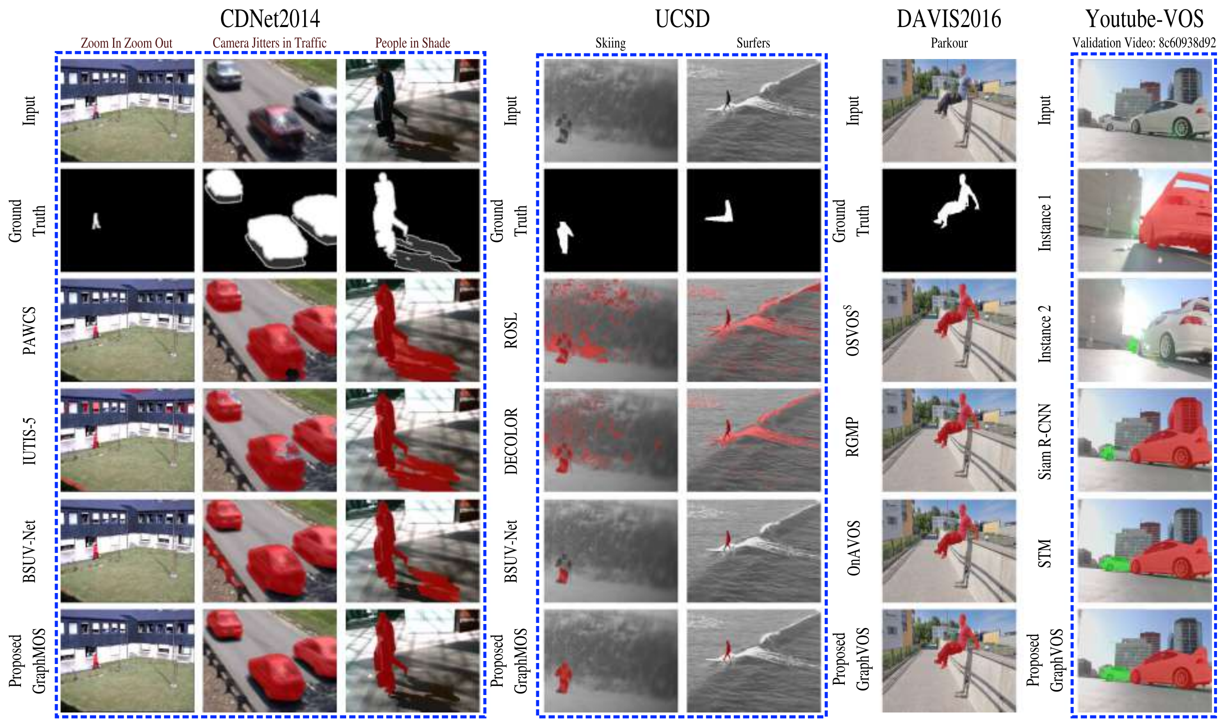


Fig. 1. Comparisons of the visual results of the proposed Graph Moving Object Segmentation (GraphMOS) and Graph Video Object Segmentation (GraphVOS) algorithms with existing SOTA methods on five MOS and two VOS challenging video sequences taken from CDNet2014 [7], UCSD [14], DAVIS2016 [15], and Youtube-VOS [16] datasets. The compared methods for MOS are: PAWCS [17], IUTIS-5 [18], BSUV-Net [19], ROSL [20], and DECOLOR [21]; while the compared methods for VOS are: OSVOS^S [22], RGMP [23], OnAVOS [24], STM [25], and Siam R-CNN [26]. Our proposed algorithms perform significantly better than the compared methods in these challenging sequences.

vision and machine learning applications such as object tracking [30], [31], MOS [32], [33], [34], and tissue community detection [35], [36]. Graphs can model data structure lying on complex irregular manifolds [37]. These methods capture the intrinsic geometric structure in data and can model data points and complex interactions among them. Each node or vertex on the graph represents one data point to which a label can be associated, and a graph can be formed by connecting vertices with edge weights that are assigned based on distance values among the data points in the feature space. Social, financial, ecological networks, and the human brain are few examples of such data structure that can be modeled on graphs [37], [38]. Graph Signal Processing (GSP) enables different types of learning and filtering operations on values associated with graph nodes [39], [40], [41], [42]. For inference, these graph models are used to classify graph signals. GSP extends the concepts of classical digital signal processing to signals supported on graphs [37], [38]. A graph signal is determined as a function over the vertices of a graph. Sampling and recovery of graph signals are essential tasks in GSP [39], [40], [41], [42]. For example, semi-supervised learning can be modeled as the reconstruction of a graph signal from its samples [40]. When data labels are presented as signals on a graph, graph signal regularization techniques can be used in the process of estimating the unknown labels for graph nodes classification [40], [41].

In the current work, we pose the problem of MOS as a semi-supervised learning problem on graphs. The nodes in the graph represent the homogeneous regions (known as superpixels) of the video sequence, and the task is to classify each homogeneous region to either a background (static component) or a moving object (foreground component) node by using the concepts of sampling and reconstruction of graph signals. Our algorithm thus lies in between the unsupervised and supervised techniques, leading to a new branch of MOS algorithms. Our proposed algorithm explores a somewhat radical departure from prior work in MOS, inspired by the theory of GSP [37], [38]. We name the proposed semi-supervised learning algorithm as Graph Moving Object Segmentation (GraphMOS), where grouped regions in the videos are modeled as nodes of a graph embedded in a high dimensional space, and a graph signal is related to the class static or moving object. Our proposed GraphMOS algorithm is composed of superpixel segmentation, background initialization, feature extraction for nodes representation, construction of a graph, sampling of graph signals, and finally, a recovery method is applied to reconstruct the graph signal from its samples. The task of the reconstruction algorithm is to classify the graph nodes. Moreover, the bandwidth of the graph signal associated with the problem shows an upper bound for the sample complexity required in semi-supervised learning [43], assuming bandlimitedness and no noise in the graph signal. Our proposed algorithm is also adapted to perform the semi-supervised Video Object Segmentation (VOS) task. The VOS is a slightly different problem than MOS, where a complete mask of one or more instances is given in the first frame, and the algorithm predicts the object masks in the subsequent frames using the first frame as the only source of information. We perform the VOS task by extending the graph signals to several categories, including background and annotated instances. We dub our proposed algorithm as GraphVOS for the VOS task. Finally, several configurations of the proposed algorithm are evaluated for the MOS and

VOS tasks outperforming many SOTA methods on the Change Detection 2014 (CDNet2014) [7], I2R [44], Scene Background Initialization (SBI2015) [13], UCSD background subtraction [14], DAVIS2016 [15], and Youtube-VOS [16] datasets.

In the current work, we extend and improve our preliminary study [45] by providing a more detailed theoretical explanation, as well as an exhaustive experimental evaluation and in-depth analyses of new results. The advantages of our algorithms are: 1) their good performances even when the background scene rapidly changes, which is difficult to handle using existing MOS and VOS methods (Fig. 1), and 2) their theoretical foundations, unlike other SOTA methods. The main contributions of the current work are summarized as follows:

- The MOS problem is posed as a graph learning problem by using the concepts of GSP. To the best of our knowledge, this is the first work that exploits the sampling and reconstruction of graph signals for MOS and VOS.
- Two theoretical developments are introduced, showing the upper-bound for the sample complexity required in semi-supervised learning under some prior assumptions in Corollary 2, as well as two bounds for the condition number of a perturbed matrix in Theorem 3.
- Extensive evaluations are performed on six publicly available MOS and VOS benchmark datasets, and we compared our algorithms with 36 existing SOTA methods with rigorous analysis. Unlike previous methods in the literature, our proposed algorithms, GraphMOS and GraphVOS, can be applied to MOS with static and moving camera sequences, as well as to the VOS task.

The rest of the paper is organized as follows. Section 2 presents related works in GSP and MOS. Section 3 explains the basic concepts and the proposed algorithms of the current work. Section 4 introduces the experimental framework. Finally, Sections 5 and 6 present the results and conclusions, respectively.

2 Related Works

This section presents brief reviews for 1) GSP and its application to computer vision, and 2) unsupervised and supervised MOS algorithms.

2.1 Graph Signal Processing

Even though the study of graphs is an ancient field, Sandryhaila and Moura were the first authors to introduce the term of discrete signal processing on graphs and later coined with the name of GSP [46]. Graph signal processing emerged with the idea of developing tools to analyze data living in irregular and complex structures [37]. From one point of view, the first developments of GSP come from the studies of low-dimensional representations for high-dimensional data through spectral graph theory, and the graph Laplacian [47]. From another perspective, several authors developed compression schemes, wavelet decomposition, filter banks on graphs, regression algorithms, and denoising using the graph Laplacian motivated by the data collected from sensor networks [48], [49], [50], [51], [52].

GSP has also been widely used in image processing and computer graphics. For example, Shi and Malik represented images as graphs to treat segmentation as a graph partitioning problem [53]. In the same way, image filtering techniques can be interpreted from a graph point of view [54]. Similarly in computer graphics, models like meshes can be naturally modeled with graphs to apply graph-based filtering and multi-resolution operations [55], [56].

In video processing, GSP is useful to model the spatiotemporal relationships among frames. For instance, the graph Cartesian product could be useful to process videos taking into account the spatiotemporal relationships of the pixels [57]. Interested readers can explore more details about GSP and its machine learning applications in a recent survey [37]. In our proposed algorithm, graphs are used to model the relationship among the nodes on videos where the graph signal represents the class foreground or background of the set of nodes in a dataset. Finally, the reconstruction of graph signals is applied to classify if a certain node is a moving or static object for MOS.

2.2 Moving Object Segmentation

There are many unsupervised methods in the literature to address the problem of MOS. These methods can be grouped as statistical [58], fuzzy [59], subspace learning [60], robust principal component analysis [61], [62], neural networks [63], and filtering-based [64] models. Interested readers may explore a complete review of unsupervised methods in survey papers [5], [11], and [27]. With the success of deep CNNs on a wide variety of computer vision applications [65], several studies have also been proposed for MOS [12], [66], [67], and VOS [22], [25], [68] applications.

The MOS supervised methods can be classified as basic CNN [66], multiscale CNN [69], [70], fully CNN [71], 3D CNN [67], and Generative Adversarial Networks (GANs) [72]. Some studies are also contributed to improving the loss functions during the training and analysis of the CNNs for the problem of MOS [73]. A complete review of deep learning-based MOS and VOS methods can be explored in recent survey studies [12], [15], and [16].

Although the results of many fully supervised deep learning-based methods show impressive performance for MOS task, the performance of the best FgSegNet method proposed by Lim *et al.* [28] shows that DCNNs methods are not efficient for unseen videos because of the lack of generalization capabilities as proved by Tezcan *et al.* [19]. This lack could be due to the limited amount of data available to train these deep learning methods for MOS, the lack of theoretical developments about the sample complexity required in deep learning, and the high model complexities of common deep neural networks. In the current work, we fill this lack by proposing a semi-supervised graph learning algorithm for MOS in unseen videos. The question of sample complexity is also solved by using the fundamental concepts of GSP under the assumption of bandlimitedness of the underlying graph signals in MOS. It is also evident that a single method, either supervised or unsupervised, cannot effectively handle all the MOS challenges of unseen videos [19]. Our proposed

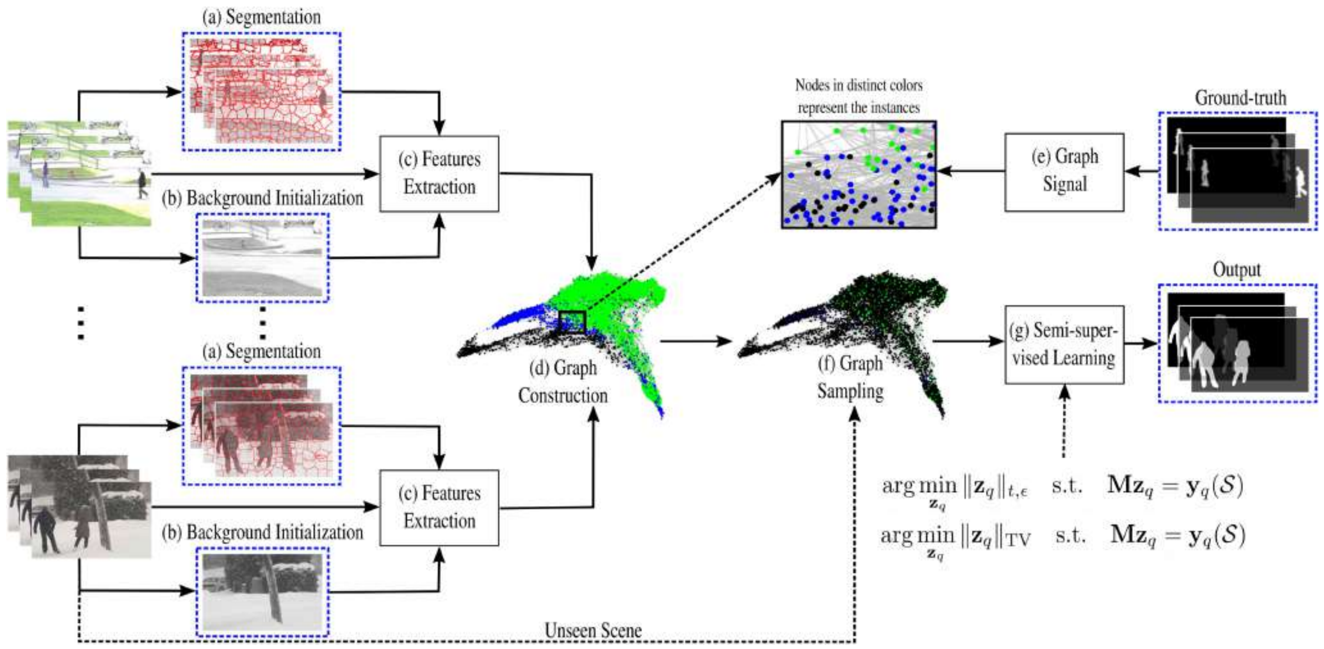

(a) Segmentation
(b) Background Initialization
(c) Features Extraction
(d) Graph Construction
Nodes in distinct colors represent the instances
(e) Graph Signal
Ground-truth
(f) Graph Sampling
(g) Semi-supervised Learning
Output
Unseen Scene
$\arg\min_{\mathbf{z}_q} \|\mathbf{z}_q\|_{t,\epsilon}$ s.t. $\mathbf{M}\mathbf{z}_q = \mathbf{y}_q(\mathcal{S})$
$\arg\min_{\mathbf{z}_q} \|\mathbf{z}_q\|_{\mathrm{TV}}$ s.t. $\mathbf{M}\mathbf{z}_q = \mathbf{y}_q(\mathcal{S})$

Fig. 2. The pipeline of the MOS algorithm with the reconstruction of graph signals. The algorithm uses background initialization and superpixel segmentation [74], [75]. Each superpixel represents a node in a graph, and the representation of each node is obtained with motion, intensity, texture, and deep features. The ground-truth is used to decide if a node is a moving (green nodes) or a static object (blue nodes). Black nodes correspond to the non-labeled images in the dataset. Finally, some nodes are sampled and the semi-supervised algorithm reconstructs all the labels in the graph.

algorithm also addresses these challenges for sequences taken from both static and moving cameras.

3 Moving Object Segmentation and Graph Signal Processing

This section presents the basic concepts of GSP and our proposed algorithms. Fig. 2 shows an overview of GraphMOS. Our proposed algorithms consist of several components including (a) superpixel segmentation, (b) background model initialization, (c) features extraction, (d) graph construction, (e) graph signal representation, (f) sampling of graph signals with an unseen scheme, and (g) a semi-supervised algorithm inspired from the theory of GSP. The novices in GSP are referred to the review papers [37], [38]. GraphMOS and GraphVOS can be viewed as three-step algorithms as follows: 1) a superpixel segmentation method [74], [76] is used to segment homogeneous regions on each frame; 2) deep and handcrafted features representation from each superpixel, including optical flow and background initialization, are used to represent the spatiotemporal information of each superpixel node; and 3) a graph reasoning algorithm, including graph construction and semi-supervised learning, is used to classify between the static and moving objects (or multiple classes in the case of GraphVOS) with few labeled samples.

3.1 Preliminaries on Graph Signals

Let $G = (\mathcal{V}, \mathcal{E})$ be an undirected and weighted graph. $\mathcal{V} = \{1, \ldots, N\}$ is the set of N nodes and $\mathcal{E} = \{(i,j)\}$ is the set of edges, where (i,j) is an edge between the vertices i and j. $\mathbf{W} \in \mathbb{R}^{N\times N}$ is the adjacency matrix of G such that $\mathbf{W}(i,j) = w_{ij} \in \mathbb{R}^+$ is the weight connecting vertices i and j. As a consequence, $\mathbf{W}$ is symmetric for undirected graphs. A graph signal is a function $y : \mathcal{V} \rightarrow \mathbb{R}$ defined on the nodes of G, and it can be represented as $\mathbf{y} \in \mathbb{R}^N$ where $\mathbf{y}(i)$ is the function evaluated on the ith node. Moreover, $\mathbf{D} \in \mathbb{R}^{N\times N}$ is the diagonal degree matrix of G such that $\mathbf{D}(i,i) = \sum_{j=1}^{N} \mathbf{W}(i,j) \ \ \forall\ i = 1, 2, \ldots, N$. Similarly, $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the positive semi-definite combinatorial Laplacian operator with eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N$ and corresponding eigenvectors $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N\}$.

The graph Fourier basis of G is defined by the spectral decomposition of $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathsf{T}}$ [37], where $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N]$ and $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N)$, where λ_i is the frequency associated to the ith eigenvalue [37]. Therefore, the Graph Fourier Transform (GFT) $\hat{\mathbf{y}}$ of the signal $\mathbf{y}$ is defined as $\hat{\mathbf{y}} = \mathbf{U}^{\mathsf{T}}\mathbf{y}$, and the inverse GFT is given by $\mathbf{y} = \mathbf{U}\hat{\mathbf{y}}$ [37].

Definition 1. *A graph signal* $\mathbf{y}$ *is called bandlimited if* $\exists\ \rho \in \{1, 2, \ldots, N-1\}$ *such that its GFT satisfies* $\hat{\mathbf{y}}(i) = 0\ \forall\ i\ >\ \rho$.

The smallest ρ that holds Definition 1 is called the bandwidth of $\mathbf{y}$. Using these notions of frequency, Pesenson [77] defined the space of all *ω-bandlimited* signals as $PW_\omega(G) = \mathrm{span}(\mathbf{U}_\rho : \lambda_\rho \leq \omega)$, where $\mathbf{U}_\rho$ represents the first ρ eigenvectors of $\mathbf{L}$, and $PW_\omega(G)$ is known as the Paley-Wiener space of G. As a consequence, a graph signal $\mathbf{y}$ has cutoff frequency ω, and bandwidth ρ if $\mathbf{y} \in PW_\omega(G)$.

3.2 Graph Nodes Representation

In our proposed algorithm, there is a need for some mechanism to represent the graph nodes. The pixels of the video frames can represent the nodes of the graph. However, computational complexity issues arise within the proposed algorithms when using pixel-level nodes. Therefore, we use a group of pixels to define each node in a graph. To that end, one common way is to decompose the input sequence into a regular block structure or superpixels. However, other object detection and segmentation methods can also be employed for node representation.

In the current work, we employ several segmentation methods including, superpixel segmentation [74], semantic segmentation [78], instance segmentation [79], block-based decomposition, and background subtraction [80] to represent the nodes in a graph, as well as to compare the performance of each of the methods. However, the aforementioned segmentation methods present a fundamental limitation in the proposed algorithm. For instance, if the segmentation methods [78], [79], and [80] do not segment moving objects, the proposed algorithm will not be able to classify background or foreground objects effectively. On the other hand, superpixel segmentation and block-based decomposition methods can process all the regions in the frames, however, these approaches may contain more graph nodes than the segmentation methods. In the later subsections, we summarize each of these methods in more detail.

3.2.1 Superpixel Segmentation

In our proposed algorithm, we use a Simple Linear Iterative Clustering (SLIC) method for superpixel segmentation [74]. SLIC adapts a k-means approach to generate a set of superpixels. The desired number of approximately equally-sized superpixels per image ζ is an important parameter in SLIC for GraphMOS and GraphVOS, because large values of ζ allow our algorithms to process more detailed regions, but may induce computational burdens on GraphMOS and GraphVOS. Several ablation studies are performed in Section 5.3 to show the performance of the proposed algorithms by varying the number of superpixels.

3.2.2 Instance Segmentation

We also test several configurations of instance segmentation methods in GraphMOS. We employ Mask Region Convolutional Neural Network (Mask R-CNN) [79], Cascade Mask R-CNN [81], Residual Networks (ResNet) [82], and ResNeSt [83] for instance segmentation. Mask R-CNN builds upon Faster R-CNN by adding a branch for predicting an object mask in parallel with the already existing Faster R-CNN network for bounding box recognition [84]. Cascade Mask R-CNN builds upon Mask R-CNN by adding a sequence of detectors [81]. Mask R-CNN contains: 1) a CNN for image feature extraction, 2) a region proposal layer, 3) Region of Interest (ROI) alignment and 4) fully connected layers in parallel with convolutional layers to perform bounding box recognition and mask prediction, respectively. The readers are referred to Appendix A in the supplementary material for further details about the instance segmentation methods, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2020.3042093.

3.2.3 Semantic Segmentation

We use the DeepLab method [78] for semantic segmentation to represent the nodes in the graph. However, our algorithms explicitly need to know the exact location of each segmented instance in a frame to represent each node in a graph. Fig. 3 shows the segmentation results of the DeepLab, Mask R-CNN, and SLIC methods on a video frame selected from the dynamic backgrounds category of the CDNet2014 dataset. GraphMOS and GraphVOS rely mainly on superpixel and instance segmentation because semantic segmentation methods do not give information about the

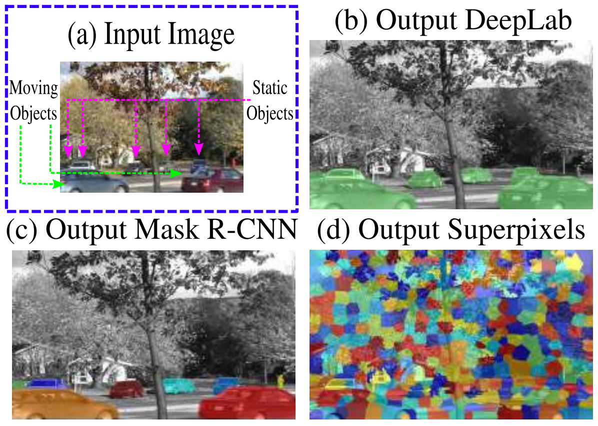


Fig. 3. Results of the semantic, instance, and superpixel segmentation using DeepLab [78], Mask R-CNN [79], and SLIC [74] methods on the sequence *fall* taken from the CDNet2014 dataset. The green-colored cars in (b), instances in different colors in (c), and homogeneous regions in (d) represent the nodes of the graph.

location of each specific instance (for example, cars as shown in Fig. 3). As a consequence, GraphMOS and GraphVOS are unable to differentiate between the parked cars in the background just behind the moving cars in the foreground when relying on semantic segmentation.

3.2.4 Other Segmentation Methods

We also decompose each video frame into non-overlapping blocks in our proposed algorithms. We use small blocks of size 8×8 to represent each node. We have also employed the background subtraction method SUBSENSE [80] to extract graph nodes. Several ablation studies are presented in Section 5.3 for comparing different segmentation methods.

3.3 Background Initialization and Feature Extraction

The MOS on unseen videos in static camera sequences can use the background of the scene as additional information. For the sake of simplicity, the temporal median filter is used as background initialization. The videos are processed in gray-scales in the current work.

The representation of the nodes is achieved with optical flow, intensity, texture, and deep features. The feature extraction module processes the ROI of the segmented regions[1] in the current frame for the current, previous, and background frames, as well as the absolute value of the difference between the current and background frames. Let $\mathbf{I}_v^t$ and $\mathbf{I}_v^{t-1}$ be the gray-scale crops corresponding to the node $v \in \mathcal{V}$ in the current (t) and previous frames ($t-1$), respectively. Let $\mathbf{B}_v$ be the crop of the background image corresponding to the node v. Let $\mathcal{P}_v$ be the set of indices corresponding to the vth segmented region. Finally, let $\mathbf{v}_x^t(\mathcal{P}_v)$ and $\mathbf{v}_y^t(\mathcal{P}_v)$ be the optical flow vectors of the current frame with support in the set of indices $\mathcal{P}_v$ for the horizontal and vertical direction, respectively. We compute the optical flow by employing the Lucas-Kanade method [85]. Fig. 4 shows the procedure to extract the features of each

1. The segmented regions are: each superpixel for SLIC, each block of pixels for block-based method, each mask for instance segmentation, distinct regions of one category for semantic segmentation, and distinct foreground regions for SuBSENSE.

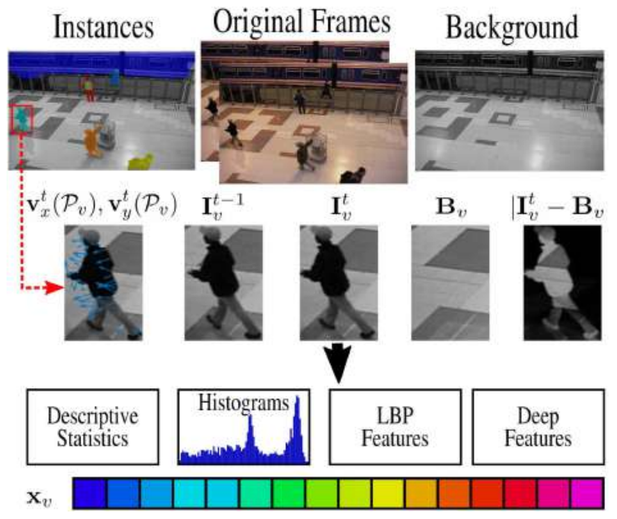


Fig. 4. Procedure to represent the nodes of the graph with a Mask R-CNN as backbone. Each mask of the segmented image represents a node in the graph, and the representation of the node is achieved with intensity, optical flow, texture, and deep features.

segmented region when the segmentation algorithm is a Mask R-CNN. The texture representation is obtained by estimating the local binary patterns [86] in $\mathbf{I}_v^t$, $\mathbf{I}_v^{t-1}$, $\mathbf{B}_v$ and $|\mathbf{I}_v^t - \mathbf{B}_v|$. The intensity histograms are computed in $\mathbf{I}_v^t(\mathcal{P}_v)$, $\mathbf{I}_v^{t-1}(\mathcal{P}_v)$, $\mathbf{B}_v(\mathcal{P}_v)$ and $|\mathbf{I}_v^t(\mathcal{P}_v) - \mathbf{B}_v(\mathcal{P}_v)|$. The vectors of orientations and magnitudes obtained from the optical flow vectors $\mathbf{v}_x^t(\mathcal{P}_v)$ and $\mathbf{v}_y^t(\mathcal{P}_v)$ are used to compute histograms and some descriptive statistics (the minimum, maximum, mean, standard deviation, mean absolute deviation, and range). Finally, the deep features of each segmented region are extracted. Inspired by the visual object tracking community [87], [88], we use a pre-trained VGG-m model [89] to extract the features from the 5th convolutional layer (Conv-5) and then a principal component analysis is applied to compress a high-dimensional feature vector into a low-dimensional vector. The representation of node v is obtained concatenating all the previous features, i.e., optical flow, intensity, texture, and deep features. Each instance is represented by a M-dimensional vector $\mathbf{x}_v$. The readers are referred to Appendix B in the supplementary material to see a detailed explanation of how the vector $\mathbf{x}_v$ is obtained, available online.

3.4 Graph Construction

The construction of the graph aims to get geometrical information from the datasets, leading to a reduction in the number of labels required in the learning process. In the current work, the construction of the graph is achieved with a k Nearest Neighborhood (k-NN) with a Gaussian kernel. Let $\mathbf{X} \in \mathbb{R}^{N \times M}$ be the matrix of N nodes, in which each node is an M-dimensional vector and $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]^\mathsf{T}$. First, the k-NN algorithm is used to connect the nodes in the graph. Afterward, vertices are connected to get an undirected and weighted graph. The weight between two connected vertices i, j is given such that $w_{ij} = \exp - \frac{d(i,j)^2}{\sigma^2}$, where $d(i,j) = \|\mathbf{x}_i - \mathbf{x}_j\|_2$, and σ^2 is the standard deviation of the Gaussian function given as

$$\sigma = \frac{1}{|\mathcal{E}| + N} \sum_{(i,j) \in \mathcal{E}} d(i,j). \tag{1}$$

As a result, the adjacency matrix $\mathbf{W}(i,j) = w_{ij} \; \forall \; (i,j) \in \mathcal{E}$ is obtained from this process.

A reasonable thought is to use a complete graph to avoid the optimization of parameter k. However, a complete graph requires a prohibitive amount of memory to store matrices $\mathbf{W}$ and $\mathbf{L}$ when dealing with huge graphs. For example, a complete graph with $N = 200000$ would require 320 gigabytes of memory to store $\mathbf{W}$, where each edge in $\mathbf{W}$ is represented with a variable of type double of 8 bytes. Other strategies for the representation of the graph can be used in our algorithm. For example, Gangapure *et al.* [90] proposed a superpixel based causal multisensor video fusion method, where the key idea is to leverage temporal information for video and then construct spatiotemporal graph models.

3.5 Graph Signal

The graph signal is a matrix $\mathbf{Y} \in \mathbb{R}^{N \times Q}$, where Q is the number of classes of the problem. $\mathbf{y}_q = \mathbf{Y}_{:,q}$ is the graph signal associated with the qth class, where $\mathbf{Y}_{:,q}$ is the qth column vector of matrix $\mathbf{Y}$. Each row of $\mathbf{Y}$ represents if certain segmented region belongs to the qth class. In the current work, the graph signal is given by the membership function $\mathbf{Y}^c$ of each class c, which takes a value of 1 on a node which belongs to the class and is 0 otherwise. For example, in MOS Q is equal to 2 that corresponds to the classes *static object* [1, 0] and *moving object* [0, 1]. The decision of whether a node is a static or moving object is based on a comparison between the ground-truth and the segmented regions of the videos. This work uses the metrics intersection over union and intersection over node to determine the foreground and background nodes.

Let $\mathcal{F}^t = \{1, \ldots, \gamma\}$ be the set of distinct regions of the foreground in the ground-truth of the current frame (t). Let $\mathcal{GT}^t$ and $\mathcal{P}_v$ be the set of indices of the ground-truth, and the set of indices of each output of the segmentation algorithm with $v \in \mathcal{V}$ associated with the current frame (t), respectively. The subset of nodes in $\mathcal{V}$ associated with the current image (t) is given by the segmented regions that exist in such a frame. The intersection over node is defined as $\xi_v = |\mathcal{I}_v^t|/|\mathcal{P}_v|$, where $\mathcal{I}_v^t = \mathcal{GT}^t \cap \mathcal{P}_v$, i.e., each node $v \in \mathcal{V}$ has an associated ξ_v. In the same way, let $\mathcal{GT}_f^t$ be the set of indices of each isolated region of the ground-truth with $f \in \mathcal{F}^t$, i.e., $\mathcal{GT}_f^t \subseteq \mathcal{GT}^t$, and $\mathcal{GT}^t = \mathcal{GT}_1^t \cup \ldots \cup \mathcal{GT}_\gamma^t$. The intersection over union is defined as $\mathbf{u}_v(f) = |\mathcal{I}_f|/|\mathcal{U}_f| \; \forall \; f \in \mathcal{F}^t$, where $\mathcal{U}_f = \mathcal{GT}_f^t \cup \mathcal{P}_v$, and $\mathcal{I}_f = \mathcal{GT}_f^t \cap \mathcal{P}_v$, i.e., each node $v \in \mathcal{V}$ has γ associated intersection over union numbers represented by the vector $\mathbf{u}_v$. Finally, $\mu_v = \max_f(\mathbf{u}(f))$ is the metric associated with intersection over union of the node v. Fig. 5 shows an example of the elements $\mathcal{GT}^t$, $\mathcal{I}_v^t$, $\mathcal{GT}_f^t$, $\mathcal{I}_f$, and $\mathcal{P}_v$ described before for the node v when the segmentation algorithm is a Mask R-CNN. In Fig. 5, for example, $\mathbf{u}_v(1) = 0 \because \mathcal{I}_1 = \emptyset \rightarrow \mu_v = \mathbf{u}_v(2)$.

Trivially, one can say that either if $\mathcal{F} \in \emptyset$, or $\mu = 0$, or $\xi = 0$, their corresponding nodes belong to the class static object, i.e., $\mathbf{Y}_{v,:} = [1, 0]$. The other cases are decided based on the segmentation method. The nodes corresponding to moving objects are determined when $\mu_v > 0.25$, or when $\xi_v > 0.45$ and $\mu_v > 0.05$, or when $\xi_v > 0.9$ and $\mu_v > 0.02$ for the background subtraction, instance, and semantic segmentation methods. On the other hand, the moving objects are determined when $\mu_v > 0.25$, or when $\xi_v > 0.45$ and $\mu_v >$

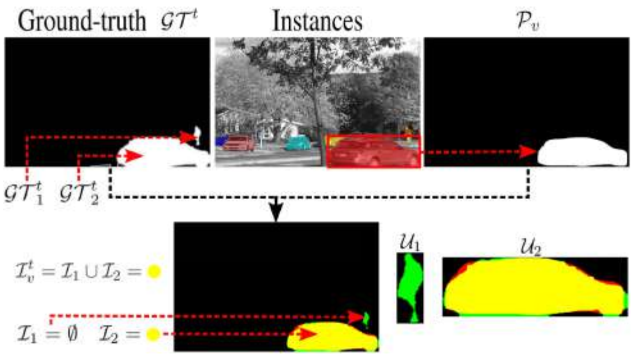

Ground-truth $\mathcal{GT}^t$
Instances
$\mathcal{P}_v$
$\mathcal{GT}_1^t$ $\mathcal{GT}_2^t$
$\mathcal{I}_v^t = \mathcal{I}_1 \cup \mathcal{I}_2 =$
$\mathcal{I}_1 = \emptyset$ $\mathcal{I}_2 =$
$\mathcal{U}_1$
$\mathcal{U}_2$

Fig. 5. Example of the sets involved in the construction of the graph signal when the segmentation algorithm is a Mask R-CNN. Yellow pixels are $\mathcal{GT}^t \cap \mathcal{P}_v$, red pixels are $\mathcal{P}_v - \mathcal{GT}^t$, and the green pixels are $\mathcal{GT}^t - \mathcal{P}_v$.

0.05, or when $\xi_v > 0.4$ for superpixel and block-based segmentation. If some node does not hold any of the conditions before, is classified as a static object. The parameters for this procedure were found empirically. Those parameters can be learned by some regression method by looking at each instance and its intersection with the ground-truth. However, the fine-tuning of those parameters is outside the scope of the current work.

3.6 Sampling of Graph Signals and Sample Complexity for Semi-Supervised Learning

Given the graph signals related to the problem of MOS and the notions of bandlimitedness in terms of $PW_\omega(G)$ from Section 3.1, the next logical step is to find a bound for the minimum number of samples required to reach perfect recovery of $\mathbf{y} \in PW_\omega(G)$. Put differently, what is the minimum amount of labeled nodes required to have a perfect classification in semi-supervised learning, given the prior assumption that the labels of the nodes are in the Paley-Wiener space of the graph? The answer is that one needs at least ρ (bandwidth) labeled nodes to achieve perfect classification. This also holds for regression of graph signals, i.e., given $\mathbf{y} \in PW_\omega(G)$, the number of sampled nodes required to get perfect reconstruction is ρ. Intuitively, a graph signal $\mathbf{y}$ is smooth in G when $\mathbf{y} \in PW_\omega(G)$. For example, suppose a sensor network of temperatures in a specific region. One would expect that the temperature of two or more nearby localities should be similar, i.e., the value of the graph signal evaluated in two or more strongly connected nodes should not be very different. As a consequence, probably one just needs the temperature of some of these nodes to reconstruct the whole graph signal in the other vertices.

To add mathematical precision to the notion of perfect reconstruction, the sampling of a graph signal is defined in terms of a subset of nodes $\mathcal{S} \subset \mathcal{V}$ with $\mathcal{S} = \{s_1, s_2, \ldots, s_m\}$, where $m = |\mathcal{S}| \leq N$ is the number of sampled nodes. The sampled graph signal is defined as $\mathbf{y}(\mathcal{S}) = \mathbf{M}\mathbf{y}$, where $\mathbf{M}$ is a binary decimation matrix whose entries are given by $\mathbf{M} = [\boldsymbol{\delta}_{s_1}, \ldots, \boldsymbol{\delta}_{s_m}]^\mathsf{T}$ and $\boldsymbol{\delta}_v$ is the $N-$dimensional Kronecker column-vector centered at v. The recovery of a graph signal from its samples $\mathbf{y}(\mathcal{S})$ can be represented as $\tilde{\mathbf{y}} = \Phi\mathbf{M}\mathbf{y}$, where $\mathbf{\Phi} \in \mathbb{R}^{N \times m}$ is an interpolation matrix. Perfect recovery is achievable if $\Phi\mathbf{M} = \mathbf{I}$, i.e., $\tilde{\mathbf{y}} = \mathbf{I}\mathbf{y} = \mathbf{y}$. Since $\text{rank}(\Phi\mathbf{M}) \leq m \leq N$, perfect reconstruction is not possible in general. However, perfect reconstruction from a sampled graph signal $\mathbf{y}(\mathcal{S})$ is possible when the sampling size $|\mathcal{S}| \geq \rho$ [39].

Theorem 1 (Chen's theorem [39]). *Let* $\mathbf{M}$ *satisfy* $\text{rank}(\mathbf{M}\mathbf{U}_\rho) = \rho$. *For all* $\mathbf{y} \in PW_\omega(G)$, *perfect recovery, i.e.,* $\mathbf{y} = \Phi\mathbf{M}\mathbf{y}$, *is achieved by choosing*

$$\mathbf{\Phi} = \mathbf{U}_\rho\mathbf{V}, \tag{2}$$

with $\mathbf{V}\mathbf{M}\mathbf{U}_\rho$ *a* $\rho \times \rho$ *identity matrix.*

Proof. see [39]. □

Theorem 1 states that perfect reconstruction of graph signal from its samples is possible when $\mathbf{y}$ lies in $PW_\omega(G)$, and the number of samples is at least ρ. Then, perfect reconstruction is achieved by choosing the interpolation operator as in Eqn. (2). A common approach to obtain a reconstructed version of $\mathbf{y}$ is given by

$$\underset{\mathbf{z}\in\text{span}(\mathbf{U}_\rho)}{arg\,min} \; ||\mathbf{M}\mathbf{z} - \mathbf{y}(\mathcal{S})||_2^2 = \mathbf{U}_\rho(\mathbf{M}\mathbf{U}_\rho)^\dagger\mathbf{y}(\mathcal{S}), \tag{3}$$

where $\mathbf{U}_\rho = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_\rho]$ is the matrix formed of the first ρ graph's eigenvectors, and $(\mathbf{M}\mathbf{U}_\rho)^\dagger$ is the pseudo-inverse of $(\mathbf{M}\mathbf{U}_\rho)$. In other words, the interpolation operator is such that $\mathbf{\Phi} = \mathbf{U}_\rho(\mathbf{M}\mathbf{U}_\rho)^\dagger$. The computation of the Laplacian eigenvectors in Eqns. (2) and (3) is computationally prohibitive for large graphs (as the ones treated in this work). In this work, the computation of $\mathbf{U}$ is avoided, for further details please see Sections 3.7, 3.8, and 3.9.

One can relate Theorem 1 to the sample complexity in semi-supervised learning as follows:

Corollary 2. *Let* $\mathbf{Y} \in \mathbb{R}^{N \times Q}$ *be a graph signal associated with a semi-supervised learning problem, where* Q *is the number of classes; and let* N_s *be the sample complexity of the semi-supervised learning problem.* $\mathbf{Y}_{i,:} = \boldsymbol{\delta}_q^\mathsf{T}$, *where* $\boldsymbol{\delta}_q$ *is a* Q*-dimensional Kronecker column vector centered at* q, *and* $\mathbf{Y}_{i,:}$ *is the ith row of* $\mathbf{Y}$. $\mathbf{Y}$ *has a set of cutoff frequencies* $\{\omega_1, \ldots, \omega_q\}$, *with corresponding bandwidths* $\{\rho_1, \ldots, \rho_q\}$ *for each graph signal* $\mathbf{Y}_{:,q}$ $\forall\ 1 \leq q \leq Q$. *Then,* N_s *is bounded such that*

$$N_s \leq \max\{\rho_1, \ldots, \rho_q\}. \tag{4}$$

Proof. From Theorem 1, one can get a perfect reconstruction of each graph signal $\mathbf{Y}_{:,i} \in PW_{\omega_i}(G)$ $\forall\ 1 \leq i \leq q$ using Eqn. (3) with at least ρ_i samples. As a consequence, the worst-case scenario in the sample complexity N_s for perfect reconstruction of $\mathbf{Y}$ is $\max\{\rho_1, \ldots, \rho_q\}$. □

3.7 Smooth Graph Signals

The prior assumption of this work relies on the bandlimitedness of the graph signals associated with the problem of MOS. This notion is also related to the smoothness of $\mathbf{y}$. Bandlimitedness was formalized in terms of the Paley-Wiener space of graph signals in Sections 3.1 and 3.6. When $\mathbf{y} \in PW_\omega(G)$, the variation of $\mathbf{y}$ is smooth in the vertex domain. From Definition 1, one knows that the GFT is required to check if $\mathbf{y} \in PW_\omega(G)$. However, the computation of the GFT requires the calculation of the eigenbasis, which is computationally prohibitive for large graphs. In the current work, the computation of $\mathbf{U}$ is avoided by leveraging notions of global smoothness in G.

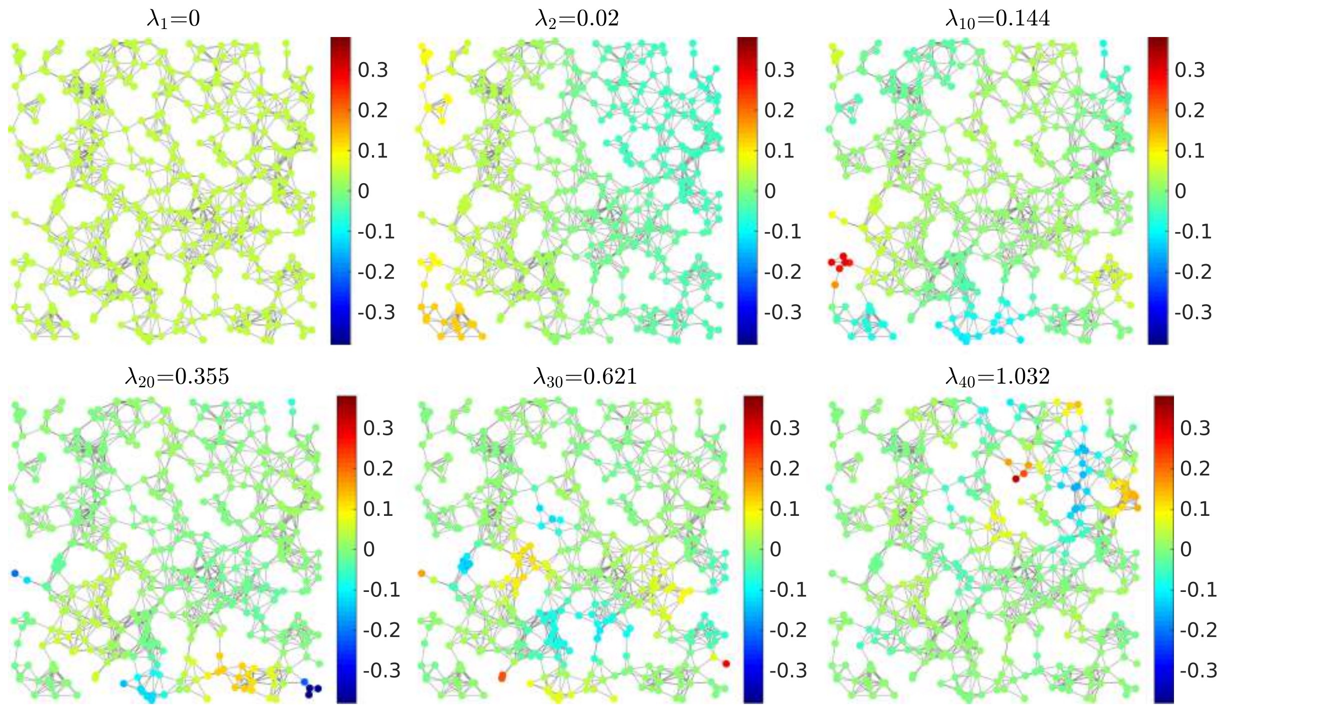


Fig. 6. Example of elementary frequencies obtained from the Laplacian matrix on a sensor network of $N = 500$. Each graph shows a frequency λ_i with its corresponding eigenvector. The lowest frequency is $\lambda_1 = 0$, corresponding to a constant graph signal, i.e., the Laplacian quadratic form of eigenvector $\mathbf{u}_1$ is given such that $\mathbf{u}_1^\mathsf{T}\mathbf{L}\mathbf{u}_1 = \lambda_1 = 0$.

Formally, notions of smoothness in $\mathbf{y}$ were introduced through concepts of *local variation* and the discrete p-Dirichlet form [38]. The local variation of $\mathbf{y}$ at vertex i is defined as

$$\|\nabla_i \mathbf{y}\|_2 \triangleq \left[\sum_{j\in\mathcal{N}_i} \mathbf{W}(i,j)[\mathbf{y}(j) - \mathbf{y}(i)]^2\right]^{\frac{1}{2}}, \tag{5}$$

where $\mathcal{N}_i$ is the set of vertices connected to the node i by one edge. Thus, the discrete p-Dirichlet form is defined as $S_p(\mathbf{y}) \triangleq \frac{1}{p}\sum_{i\in\mathcal{V}} \|\nabla_i \mathbf{y}\|_2^p$, then

$$S_p(\mathbf{y}) = \frac{1}{p}\sum_{i\in\mathcal{V}}\left[\sum_{j\in\mathcal{N}_i} \mathbf{W}(i,j)[\mathbf{y}(j) - \mathbf{y}(i)]^2\right]^{\frac{p}{2}}. \tag{6}$$

For example, when $p = 2$

$$S_2(\mathbf{y}) = \sum_{(i,j)\in\mathcal{E}} \mathbf{W}(i,j)[\mathbf{y}(j) - \mathbf{y}(i)]^2 = \mathbf{y}^\mathsf{T}\mathbf{L}\mathbf{y}. \tag{7}$$

$S_2(\mathbf{y})$ is known as the graph Laplacian quadratic form [38]. Notice that $S_2(\mathbf{y}) = 0 \Leftrightarrow \mathbf{y} = \tau\mathbf{1}$, where τ is a constant; and more generally, $S_2(\mathbf{y})$ is small when the graph signal $\mathbf{y}$ has similar values at neighboring nodes connected by an edge, i.e., when the signal is smooth.

The Laplacian quadratic form in Eqn. (7) has been used as regularizer in reconstruction of graph signals, and semi-supervised learning problems [91], where this regularizer looks for smooth graph signals. Intuitively, there is a relationship between the smoothness of a graph signal and its bandwidth. For example, the variational problem [92] leads to the same solution of Eqn. (3) when $\mathbf{y}$ holds Definition 1 [43] (for further details see Section 3.8). Therefore, the minimization of the p-Dirichlet form is aligned with the prior assumption of bandlimitedness, without the explicit computation of the eigenbasis $\mathbf{U}$ and the bandwidth ρ of $\mathbf{y}$. Formally, since $\mathbf{L}$ is positive semi-definite for undirected graphs, all the eigenvalues are non-negative and real, and a full set of orthogonal eigenvectors can be obtained as explained in Section 3.1. The matrix of eigenvectors $\mathbf{U}$ is known as the GFT matrix of the graph. The eigenvalue-eigenvector pairs can be viewed as successive optimizers of the Rayleigh quotient, where the ith pair $\lambda_i, \mathbf{u}_i$ solves

$$\mathbf{u}_i = \underset{\mathbf{y}^\mathsf{T}\mathbf{u}_{i'}=0, i'=0,\ldots,i-1}{\arg\min} \frac{\mathbf{y}^\mathsf{T}\mathbf{L}\mathbf{y}}{\mathbf{y}^\mathsf{T}\mathbf{y}}, \tag{8}$$

with $\lambda_i = \mathbf{u}_i^\mathsf{T}\mathbf{L}\mathbf{u}_i$ if $\mathbf{u}_i^\mathsf{T}\mathbf{u}_i = 1$. The term $\mathbf{y}^\mathsf{T}\mathbf{L}\mathbf{y}$ is precisely the Laplacian quadratic form of the graph signal $\mathbf{y}$ [38], i.e., the GFT provides an orthogonal basis with increased variation [37]. Fig. 6 shows an example of the eigenvectors of a weighted undirected sensor network with 500 nodes. One can notice that the eigenvector $\mathbf{u}_i$ has more variations as the value of λ_i increases, where the Laplacian quadratic form of $\mathbf{u}_1$, given as $\mathbf{u}_1^\mathsf{T}\mathbf{L}\mathbf{u}_1 = \lambda_1 = 0$, corresponds to a constant-valued eigenvector. Since the GFT of a graph signal is given such that $\hat{\mathbf{y}} = \mathbf{U}^\mathsf{T}\mathbf{y}$, one can precisely represent a bandlimited graph signal in the Fourier domain as $\hat{\mathbf{y}} = [\mathbf{u}_1, \ldots, \mathbf{u}_\rho, \mathbf{0}, \ldots, 0]^\mathsf{T}\mathbf{y}$ according to Definition 1, i.e., a mapping of the first ρ eigenvectors. As a consequence, a bandlimited graph signal is smooth on undirected graphs.

In semi-supervised learning, the number of samples required to get perfect classification increases when the bandwidth ρ increases as expected from Corollary 2. In practice, graph signals are only approximately bandlimited [40]. As a consequence, the classification error is bounded by a value ϕ [43].

3.8 Minimization of the Sobolev Norm

One of the semi-supervised learning methods in this work is based on the variational splines of Pesenson [92].

Definition 2. *Let $\mathbf{g}^*(v)$ be the complex conjugate of $\mathbf{g}(v)$. The space $L_2(G)$ is the Hilbert space of all complex-valued functions $\mathbf{f}: \mathcal{V} \rightarrow \mathbb{C}$ in the graph G with the inner product $\langle \mathbf{f}, \mathbf{g} \rangle = \sum_{v \in \mathcal{V}} \mathbf{f}(v)\mathbf{g}^*(v)\mathbf{D}(v,v)$, and $\|\mathbf{f}\| = \|\mathbf{f}\|_0 = (\sum_{v \in \mathcal{V}} |\mathbf{f}(v)|^2 \mathbf{D}(v,v))^{1/2}$*

Definition 3. *For a fixed $\epsilon \geq 0$ the Sobolev norm is introduced by the following formula:*

$$\|\mathbf{f}\|_{\alpha,\epsilon} = \|(\mathbf{L} + \epsilon\mathbf{I})^{\alpha/2}\mathbf{f}\|_2^2, \alpha \in \mathbb{R}. \tag{9}$$

Given a graph signal of sampled labels $\mathbf{y}_q(\mathcal{S}) = \mathbf{M}\mathbf{y}_q$, a positive $\alpha > 0$, and a non-negative $\epsilon \geq 0$, the variational problem for semi-supervised learning is stated as follows: find a vector $\mathbf{z}_q$ from the space $L_2(G)$ with the following properties: $\mathbf{z}_q(\mathcal{S}) = \mathbf{M}\mathbf{z}_q = \mathbf{y}_q(\mathcal{S})$, and $\mathbf{z}_q$ minimizes functional $\mathbf{z}_q \rightarrow \|(\mathbf{L} + \epsilon\mathbf{I})^{\alpha/2}\mathbf{z}_q\|_2$. In other words, the variational problem is trying to solve the following optimization problem:

$$\begin{aligned} &\underset{\mathbf{z}_q}{\arg\min} \|\mathbf{z}_q\|_{\alpha,\epsilon} \quad \text{s.t.} \quad \mathbf{M}\mathbf{z}_q = \mathbf{y}_q(\mathcal{S}) \rightarrow \\ &\underset{\mathbf{z}_q}{\arg\min}\, \mathbf{z}_q^{\mathsf{T}}(\mathbf{L} + \epsilon\mathbf{I})^{\alpha}\mathbf{z}_q \quad \text{s.t.} \quad \mathbf{M}\mathbf{z}_q - \mathbf{y}_q(\mathcal{S}) = 0. \end{aligned} \tag{10}$$

Equation (10) is a convex optimization problem since the term $\mathbf{z}_q^{\mathsf{T}}(\mathbf{L} + \epsilon\mathbf{I})^{\alpha}\mathbf{z}_q$ is a quadratic convex function in $\mathbf{z}_q$; and the term $\mathbf{M}\mathbf{z}_q - \mathbf{y}_q(\mathcal{S})$ is affine in $\mathbf{z}_q$. The semi-supervised learning problem is solved by determining the solution of Eqn. (10) for $q = 1, \ldots, Q$.

The minimization of the Sobolev norm in Eqn. (10) is closely related to the Laplacian quadratic form. The Laplacian quadratic form of $\mathbf{y}$ is given by $\mathbf{y}^{\mathsf{T}}\mathbf{L}\mathbf{y}$ as shown in Eqn. (7). In the same way, $\mathbf{y}^{\mathsf{T}}\mathbf{L}^{\alpha}\mathbf{y}$ is known as the empirical iterated Laplacian regularizer or *higher-order regularization* [93] and has been used for regression and classification tasks in semi-supervised learning [91]. On the other hand, the Sobolev norm is such that $\mathbf{y}^{\mathsf{T}}(\mathbf{L} + \epsilon\mathbf{I})^{\alpha}\mathbf{y}$. The Sobolev norm of $\mathbf{y}$ is the higher-order regularization with an addition of the semi-definite perturbation matrix $\epsilon\mathbf{I}$. For $\epsilon > 0$, $(\mathbf{L} + \epsilon\mathbf{I})$ is always invertible even though $\det(\mathbf{L}) = 0$ for undirected and connected graphs. Intuitively, the value ϵ is related to the stability of the inverse of $(\mathbf{L} + \epsilon\mathbf{I})$.

Theorem 3. *Given an undirected, connected graph G with combinatorial Laplacian matrix $\mathbf{L}$ such that $\operatorname{rank}(\mathbf{L}) = N - 1$, and let $\boldsymbol{\Psi} \in \mathbb{R}^{N \times N}$ a perturbation matrix. The summation $\mathbf{L} + \boldsymbol{\Psi}$ has a lower and upper bound in the condition number in the ℓ_2-norm such that*

$$\frac{\sigma_{\max}(\mathbf{L} + \boldsymbol{\Psi})}{\sigma_{\max}(\boldsymbol{\Psi})} \leq \kappa(\mathbf{L} + \boldsymbol{\Psi}) \leq \frac{\sigma_{\max}(\mathbf{L}) + \sigma_{\max}(\boldsymbol{\Psi})}{\sigma_{\min}(\mathbf{L} + \boldsymbol{\Psi})}, \tag{11}$$

where $\kappa(\mathbf{L} + \boldsymbol{\Psi})$ is the condition number of $\mathbf{L} + \boldsymbol{\Psi}$, and $\sigma_{\max}(\cdot)$, $\sigma_{\min}(\cdot)$ represent the maximum and minimum singular values, respectively.

Proof. see Appendix C in the supplementary material, available online. □

Theorem 3 provides a lower and upper bound in the condition number[2] of $\mathbf{L} + \boldsymbol{\Psi}$. The lower bound can be achieved by computing the GFT of $\mathbf{L}$. The addition of a perturbation matrix to the Laplacian matrix is implicitly changing the eigenvalues of $\mathbf{L}$, however, Theorem 3 does not state how the eigenvalues of $\mathbf{L}$ change. In matrix theory, Weyl's inequality [94] is a theorem about how the eigenvalues of a perturbed matrix change.

Theorem 4 (Weyl's Theorem [94]). *Let $\mathbf{L}$ and $\boldsymbol{\Psi}$ be Hermitian matrices with set of eigenvalues $\{\lambda_1, \lambda_2, \ldots, \lambda_N\}$ and $\{\psi_1, \psi_2, \ldots, \psi_N\}$, respectively. The matrix $\mathbf{L}_{\text{per}} = \mathbf{L} + \boldsymbol{\Psi}$ has a set of eigenvalues $\{\nu_1, \nu_2, \ldots, \nu_N\}$ where the inequalities $\lambda_i + \psi_1 \leq \nu_i \leq \lambda_i + \psi_N$ hold for $i = 1, 2, \ldots, N$.*

Proof: see [94].

In Theorem 4, if $\boldsymbol{\Psi} \succ \mathbf{0}$, i.e., $\psi_i > 0 \; \forall \; 1 \leq i \leq N$, then this implies that $\nu_i > \lambda_i \; \forall \; i = 1, 2, \ldots, N$. Weyl's Theorem indicates how the eigenvalues of $\mathbf{L}$ change after adding a perturbation matrix, and it gives insights about the structure of $\boldsymbol{\Psi}$. It is desirable to have $\det(\mathbf{L}_{\text{per}}) \neq 0$, then $\boldsymbol{\Psi}$ should be positive definite. Assuming $\boldsymbol{\Psi} = \epsilon\mathbf{I}$, where $\epsilon \in \mathbb{R}^+$ we have $\sigma_{\max}(\boldsymbol{\Psi}) = \epsilon$. Furthermore, the minimum eigenvalue $\sigma_{\min}$ of $\mathbf{L} + \boldsymbol{\Psi}$ is strictly greater than zero according to Theorem 4, i.e., $\nu_1 > \lambda_1$, since $\sigma_{\min}(\mathbf{L} + \boldsymbol{\Psi}) > 0$ the upper bound in Eqn. (11) is

$$\kappa(\mathbf{L} + \boldsymbol{\Psi}) \leq \frac{\sigma_{\max}(\mathbf{L}) + \epsilon}{\sigma_{\min}(\mathbf{L} + \boldsymbol{\Psi})} < \infty. \tag{12}$$

The term $(\mathbf{L} + \epsilon\mathbf{I})$ is precisely the first term in the Sobolev norm in Eqn. (9), and the invertible term in the variational problem in Eqn. (10). ϵ is fundamentally related to how well conditioned is the variational problem.

Since $(\mathbf{L} + \epsilon\mathbf{I})^{\alpha}$ is always invertible for $\alpha > 0$ and $\epsilon > 0$, one can show that the semi-supervised learning problem in Eqn. (10) has a closed-form solution given by

$$((\mathbf{L} + \epsilon\mathbf{I})^{-1})^{\alpha}\mathbf{M}^{\mathsf{T}}(\mathbf{M}((\mathbf{L} + \epsilon\mathbf{I})^{-1})^{\alpha}\mathbf{M}^{\mathsf{T}})^{-1}\mathbf{Y}(\mathcal{S}), \tag{13}$$

where $\mathbf{Y}(\mathcal{S})$ is the sub-matrix of $\mathbf{Y}$ with rows indexed by $\mathcal{S}$. The proof is shown in Appendix D in the supplementary material, available online. For small values of N, the minimization of the Sobolev norm can be solved with the closed-form solution in Eqn. (13). For larger values of N, this minimization can be achieved with iterative methods such as the interior-point method from quadratic programming.

3.9 Minimization of the Total Variation

Another reconstruction algorithm for semi-supervised learning in this work is based on the Total Variation (TV) of graph signals. TV of graph signals is defined as [95]

$$\|\mathbf{y}\|_{\text{TV}} = \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{N}_i} \sqrt{\mathbf{W}(i,j)}\|\mathbf{y}(j) - \mathbf{y}(i)\|_1. \tag{14}$$

The minimization of the TV of $\mathbf{y}$ is related to the cluster assumption [96], and end ups in piecewise constant signals. The semi-supervised algorithm solves the following optimization problem:

2. The condition number $\kappa(\mathbf{A})$ associated with the square matrix $\mathbf{A}$ is a measure of how well or ill conditioned is the inversion of $\mathbf{A}$.

$$\arg\min_{\mathbf{z}_q} \|\mathbf{z}_q\|_{\mathrm{TV}} \quad \text{s.t.} \quad \mathbf{M}\mathbf{z}_q = \mathbf{y}_q(\mathcal{S}), \tag{15}$$

for $q = 1, \ldots, Q$. Basically, Eqn. (15) is minimizing the TV of the reconstructed graph signal such that $\mathbf{M}\mathbf{z}_q = \mathbf{y}_q(\mathcal{S})$. The minimization of the TV in Eqn. (15) involves a non-differentiable objective function, which discards any gradient descent method. In the current work, the minimization of the TV is solved with a primal dual approach [97]. Let $\mathbf{P} \in \mathbb{R}^{|\mathcal{E}|\times N}$ be the *incidence matrix* of G defined as

$$\mathbf{P}(e,i) = \begin{cases} \sqrt{\mathbf{W}(e)} & \text{if } i = \min\{i,j\} \\ -\sqrt{\mathbf{W}(e)} & \text{if } i = \max\{i,j\} \\ 0 & \text{otherwise,} \end{cases} \tag{16}$$

where $i \in \mathcal{V}$ and $e \in \mathcal{E}$ (in this case the edges are represented by the numbers $\{1, \ldots, |\mathcal{E}|\}$). The matrix $\mathbf{P}$ allows to represent the TV of the graph signal as $\|\mathbf{y}\|_{\mathrm{TV}} = \|\mathbf{P}\mathbf{y}\|_1$ [96]. As a consequence, Eqn. (15) can be formulated as the primal equivalent unconstrained convex optimization problem

$$\arg\min_{\mathbf{z}_q} g(\mathbf{P}\mathbf{z}_q) + h(\mathbf{z}_q), \tag{17}$$

$$g(\mathbf{x}) \triangleq \|\mathbf{x}\|_1, \text{ and } h(\mathbf{z}_q) \triangleq \begin{cases} \infty & \text{if } \mathbf{z}_q \notin \mathcal{Q} \\ 0 & \text{if } \mathbf{z}_q \in \mathcal{Q}, \end{cases} \tag{18}$$

where $h(\mathbf{z}_q)$ is the indicator function and $\mathcal{Q} = \{\mathbf{z}_q \in \mathbb{R}^N : \mathbf{M}\mathbf{z}_q = \mathbf{y}_q(\mathcal{S})\}$.

Definition 4. *Let $f : \mathbb{R}^n \to \mathbb{R}$. The function $f' : \mathbb{R}^n \to \mathbb{R}$ is the convex conjugate such that $f'(\mathbf{x}) \triangleq \sup_{\bar{\mathbf{x}}} \mathbf{x}^{\mathsf{T}}\bar{\mathbf{x}} - f(\bar{\mathbf{x}})$ [98].*

The dual problem associated with the minimization of the TV is given by $\arg\max_{\mathbf{x}} - h'(-\mathbf{P}^{\mathsf{T}}\mathbf{x}) - g'(\mathbf{x})$, where h' is the convex conjugate of the function h. The minimization of the TV is solved with the primal-dual approach of first order [97] using the Unlocbox toolbox [99]. For further details, the readers are referred to the references [96], [99], [100].

3.10 GraphMOS and GraphVOS in a Nutshell

GraphMOS and GraphVOS are transductive algorithms because they require that all nodes in the graph are present to solve the semi-supervised learning problem. Our algorithms are introduced with a batch method for superpixel segmentation with parameters batch size η, and ground-truth size χ, where η is the number of frames that will be processed on each iteration, and χ is the number of frames with ground-truth that will be randomly selected from other sequences (unseen scheme) to solve the semi-supervised learning problem. Algorithm 1 shows the pseudo-code for GraphMOS with batch processing. GraphMOS can also be solved using the whole dataset to construct the graph, but in this case, our algorithm will select a subset of nodes $\mathcal{S} \in \mathcal{V}$ with an unseen scheme to construct the graph signal $\mathbf{Y}$. Notice that in the case of processing the whole dataset the parameter η is not required, but one has to use instance or semantic segmentation methods to reduce the number of nodes. Also, notice that the parameters ϵ and α are not required when solving the semi-supervised learning problem with TV minimization.

Algorithm 1. Graph Moving Object Segmentation

```
  Input: Video sequences in the dataset.
  Initialization: Parameters $\zeta$, $k$, $\epsilon$, $\alpha$, $\eta$, $\chi$.
 1: Select $z$th video to be segmented
 2: while there are still frames to process in video $z$ do
 3:     Select $\eta$ frames, in order, from $z$ for batch processing
 4:     Randomly select $\chi$ frames from other videos
 5:     Compute $\zeta$ superpixels for the $\eta + \chi$ frames
 6:     Compute $\mathbf{X}$ from the $\eta + \chi$ frames
 7:     Compute $\mathbf{Y}$ from the $\chi$ ground-truth frames
 8:     Find the set of $k$-NN of $\mathbf{x}_i$ $\forall$ $i \in \mathcal{N}$
 9:     Compute $\sigma$ using Eqn. (1)
10:     for $(i,j)$ in the set of $k$-NN do
11:         $\mathbf{W}(i,j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2/\sigma^2)$
12:     end for
13:     $\mathbf{D} = \mathrm{diag}(\mathbf{W1})$, $\mathbf{L} = \mathbf{D} - \mathbf{W}$
14:     Solve either Eqn. (10) or (15) for $q = 1, \ldots, Q$
15:     Get $\mathbf{Z}$ as the solution of the semi-supervised problem
16:     Use $\mathbf{Z}$ to compute the moving objects of the $\eta$ frames
17: end while
```

3.10.1 Extension to Semi-Supervised Video Object Segmentation

Algorithm 1 works when our algorithm requires to segment moving or static objects (as in the case of the MOS datasets). However, the extension to VOS with multiple objects is straightforward using the two following steps: 1) the optical flow estimation of each video is computed backward (from the last to the first frame) since VOS datasets usually give the objects to be segmented in the first frame, if the optical flow is computed as usual, GraphVOS will not have motion information of the first frame; and 2) the matrix of graph signals $\mathbf{Y}$ should be extended with Q equal to the number of objects to be segmented for each video.

In the current work, GraphVOS is evaluated in the semi-supervised VOS task. However, this should not be confused with our semi-supervised learning approach. The VOS community decided to name semi-supervised VOS to the problem where some intermediate annotated information is given to the algorithm.

4 Experimental Framework

This section introduces the datasets used in the current work, the evaluation metrics, the experiments, and the implementation details of GraphMOS and GraphVOS.

4.1 Datasets

Our proposed algorithms are evaluated on a variety of datasets for both MOS and VOS tasks. More specifically, we evaluated the performance of the proposed GraphMOS algorithm for the sequences taken from static and moving cameras on several background modeling challenges. The main goal in GraphMOS is to segment moving objects, also known as background subtraction. While GraphVOS algorithm is evaluated on a semi-supervised VOS task where the object mask is given in the first frame, and the task is to segment the same object in the consecutive frames.

4.1.1 MOS From Static Camera

For sequences taken from static camera, the GraphMOS is evaluated on three datasets including Change Detection (CDNet2014) [7], I2R [44], and Scene Background Initialization (SBI2015) [13]. CDNet2014 is the reference dataset in the MOS community. This dataset is categorized into 11 main challenges including, bad weather, low frame rate, night videos, turbulence, baseline, dynamic backgrounds, PTZ, camera jitter, intermittent object motion, shadow, and thermal. PTZ category presents sequences taken from PTZ cameras, while the camera jitter category contains sequences of jittering effects. Each challenge contains from four up to six videos. Every video contains a certain amount of ground-truth frames, in which the ground truth shows the foreground and background. The average resolution of the sequences is 320×240.

I2R dataset contains nine challenging sequences with an average resolution of 240×192. Each sequence contains only 20 images of ground-truth. SBI2015 dataset contains 14 sequences with an average resolution of 328×246. The dataset presents the challenges of cluttered moving objects and slow object motion. This dataset was originally designed to evaluate background initialization methods. However, Wang *et al.* provided the ground-truth for MOS for SBI2015 [70].

4.1.2 MOS for Moving Camera

GraphMOS is also evaluated on the UCSD dataset [14] containing 18 challenging moving camera sequences. This dataset also presents severe dynamic background variations. Each video of UCSD is partially or fully annotated with pixel-level ground-truth images of foreground and background. The average resolution of the dataset is 230×320.

4.1.3 Semi-Supervised VOS

Our proposed GraphVOS algorithm is evaluated on DAVIS2016 [15] and Youtube-VOS [16] datasets. DAVIS2016 is one of the most popular benchmarks for single VOS in each sequence. It contains 50 fully annotated sequences with high-quality masks. The dataset is officially split into 20 validation videos and 30 training videos. The videos in this dataset contain a variety of challenges, such as background clutter, deformation, occlusion, scale variation, and shape complexity, to name a few [16].

Youtube-VOS is the largest dataset in the VOS community for multiple object segmentation. This dataset contains 4,453 videos, which are split into 3,471 videos for training, 474 for validation, and 508 for testing. The validation set includes 91 object categories, where 65 categories are present in the training set (seen objects), and 26 categories are unseen. The training and validation videos have pixel-level ground truth annotations for every 5th frame (6 fps).

4.2 Evaluation Metrics

The F-measure metric is used for the MOS task to compare the performance of our GraphMOS algorithm with SOTA methods [7]. The F-measure metric is defined as follows:

$$\text{F-measure} = 2\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \text{ where} \tag{19}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \text{ Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{20}$$

where TP, FP, and FN are the number of true positives, false positives, and false negatives pixels, respectively.

For the VOS task, we evaluate the segmentation results using the region similarity and the contour accuracy metrics described in [15]. For region similarity, the Jaccard index $\mathcal{J}$ is used to compare the SOTA methods. The $\mathcal{J}$ is defined as the intersection over union of the predicted segmentation mask and ground-truth mask. For contour accuracy $\mathcal{F}$, the same F-measure metric is used as defined in Eqn. (19). For Youtube-VOS, the results of GraphVOS have been uploaded to the online evaluation server [16], while in the case of DAVIS2016, the official toolkit is used [15].

4.3 Experiments

A thorough comparison with SOTA methods, along with a set of comprehensive ablation studies of GraphMOS and GraphVOS are performed. The ablation studies analyze some specific elements of the pipeline in Fig. 2, like the segmentation method, the number of superpixels in the case of SLIC, and the feature extraction procedure. The instance segmentation methods are performed with Mask R-CNN using ResNet-50 as the backbone, while Cascade Mask R-CNN is performed using ResNeSt-200. Further ablations are given in Appendix E in the supplementary material for the construction of the graph and the semi-supervised learning algorithm, available online.

Finally, the last experiment is devoted to testing Corollary 2. Since the calculation of $\mathbf{U}$ is computationally prohibitive for the configurations involved in the ablation studies, in this case, a graph with $N = 7443$ nodes is constructed with the sequences *backdoor*, *bungalows*, and *busStation* of the challenge shadow of CDNet2014, using a k-NN with $k = 30$, and a Mask R-CNN. In practice, graph signals are only approximately bandlimited [40], i.e., the GFT of $\mathbf{y}$ has an exponentially decaying shape. The bandwidth in this experiment is computed as the ρ where 90 percent of the spectral energy of $\mathbf{y}$ is. The spectral energy is defined as $\sum_{i \in \mathcal{V}} \hat{\mathbf{y}}^2(i)$. As a consequence of this approximation, the classification error can be bounded by some value ϕ [43]. The last experiment computes the classification error in the non-sampled nodes for the sample sizes in the set $m = \{10, 20, 30, \ldots, 400\}$, using random sampling and the Sobolev minimization with $\epsilon = 0.2$ and $\alpha = 1$. This experiment is performed with a Monte Carlo cross-validation with 200 repetitions.

4.4 Parameters Settings

Our algorithms contain several parameters such as the number of superpixels ζ, the number of k neighbors for k-NN in the graph construction, the parameters ϵ and α for the Sobolev norm, the batch size η, and the number of selected ground-truth frames χ. Table 1 shows a summary of the best parameters for GraphMOS and GraphVOS within each dataset in the current work. Several ablation studies have been performed for these parameters. See Tables 1 and 2 in the supplementary material for k and ϵ, and Table 12 for ζ, available online.

TABLE 1
Summary of the Parameters Used in Our Experiments for Each Dataset

Dataset	ζ	k	ϵ	α	η	χ
CDNet2014	200	30	0.2	1	100	350
I2R	200	30	0.2	1	100	150
SBI2015	300	30	0.2	1	100	60
UCSD	200	30	0.2	1	100	40
DAVIS2016	300	30	0.2	1	100	25
Youtube-VOS	200	30	0.2	1	100	10

4.5 Implementation Details

The instance and semantic segmentation algorithms were implemented using Pytorch and Detectron2 [101]. The algorithm for the reconstruction of graph signals was implemented using the graph signal processing toolbox [102] and the Matlab convex optimization toolbox [99]. The code has been made available.[3] For the comparison with our algorithm, most of the MOS methods were implemented with the BGSLibrary [103] and the LRSLibrary [104] with default parameters by reference. The experiments are executed on a laptop with a processor Intel Core i7 with 16 gigabytes of memory RAM. Using handcrafted features, GraphMOS algorithm with Mask R-CNN takes 3.73 FPS, with Cascaded Mask R-CNN it takes 1.76 FPS, with Deeplab method it takes 13.14 FPS, and with SuBSENSE method GraphMOS takes 15.11 FPS to segment the moving objects. In case of using deep features with block and superpixel-based segmentation methods, the experiments of GraphMOS and GraphVOS are conducted on a powerful Nvidia DGX-2 server machine. GraphMOS takes 1.42 and 2.31 FPS for MOS using superpixel and block-based graph construction.

5 Results and Discussion

Our proposed algorithms are compared with 36 SOTA methods. GraphMOS is compared with 20 SOTA methods including SWCD [105], FTSG [106], SuBSENSE [80], WeSamBE [107], PAWCS [17], WisenetMD [108], IUTIS-5 [18], SemanticBGS [109], BSUV-net [19], MoG [58], DECOLOR [21], ViBe [110], 3WD [8], GRASTA [111], FgSegNet v2 [28], ROSL [20], ADMM [112], noncvxRPCA [113], and OR1MP [114] for the MOS task. Mask R-CNN can directly solve the MOS problem. However, Mask R-CNN does not include temporal information, leading to an ill-conditioned problem. In our experiments, we have also trained Mask R-CNN directly on CDNet2014 sequences with an unseen scheme. Each Mask R-CNN is trained with 1000 epochs using learning rate of 0.00025 and a batch size of 2.

In addition, GraphVOS is compared with 16 SOTA methods including OSMN [68], MSK [115], RGMP [23], OnAVOS [24], ROVS [116], OSVOS [117], S2S [16], A-GAME [118], PReMVOS [119], MaskRNN [120], FEELVOS [121], FAVOS [122], OSVOS^S [22], DyeNet [123], Siam R-CNN [26], and STM [25] for the VOS task.

3. https://github.com/jhonygiraldo/GraphMOS

5.1 Qualitative Evaluations

Table 2 shows some of the visual results of the proposed algorithms compared with several SOTA methods on CDNet2014, UCSD, DAVIS2016, and Youtube-VOS datasets. The CDNet2014 sequences *WetSnow*, *Fall*, *IntermittentPan*, and *FluidHighway* present bad weather conditions, dynamic background variations, panning of the scene, and nighttime lighting variations challenges. GraphMOS shows the best visual results, while BSUV-Net and SuBSENSE show competitive performance for these sequences. *Birds* and *Cyclists* sequences of UCSD present highly dynamic background variations challenges, including rippling water surface and swaying of bushes. The compared methods do not handle these sequences accurately, while only GraphMOS can handle these sequences successfully.

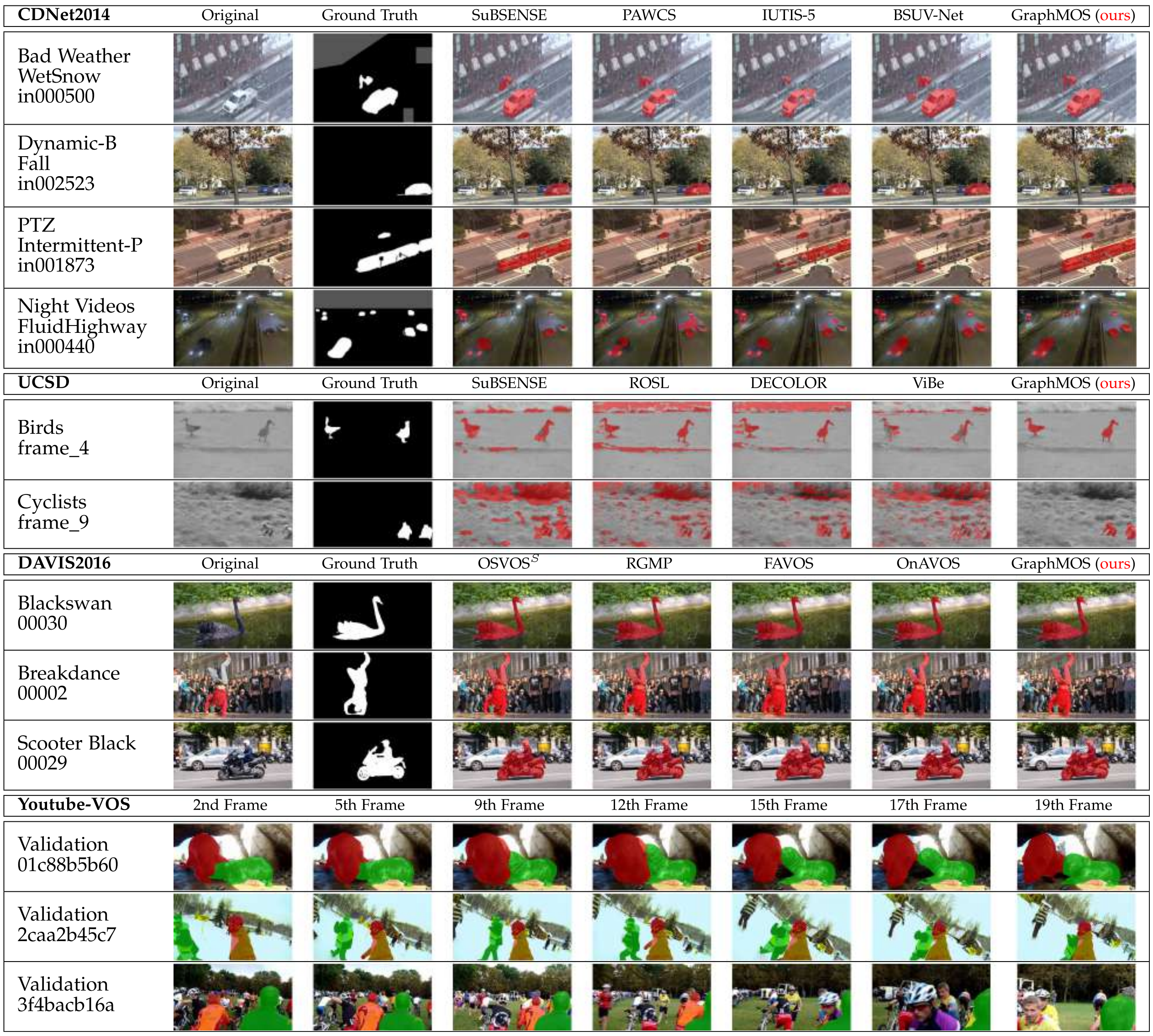
The *Blackswan*, *Breakdance*, and *Scooter Black* sequences of DAVIS2016 present the background clutter, shape deformation, and occlusion challenges. It is shown that all of the compared methods, including our proposed GraphVOS algorithm, show competitive performance in these sequences. Thus, these sequences do not put a great burden on the compared methods. In the case of YouTube-VOS sequences, some key frames of important moments (e.g., before and after occlusion and background clutter) are shown for three validation videos. GraphVOS also shows competitive performance on these sequences. The excellent performance of the proposed algorithms is because of the semi-supervised learning method for accurate classification of background or foreground graph nodes.

5.2 Quantitative Results

Tables 3, 4, 5, and 6 show the comparisons of the qualitative results of the GraphMOS algorithm on CDNet2014, I2R, SBI2015, and UCSD datasets for MOS while Tables 7 and 8 demonstrate the comparison of our proposed GraphVOS algorithm on DAVIS2016 and Youtube-VOS datasets for VOS with several SOTA methods. Online Learning (OL) in Table 7 means that the authors of those methods fine-tune the backbone networks using scribbles given by the users. However, OL is computationally too expensive to be used within an interactive tool, limiting its practical use. Within these tables, the best and second best performing methods are shown in **red** and **blue**, respectively. Our proposed algorithms show competitive performance as compared to SOTA methods on all datasets.

For the CDNet2014 dataset (Table 3), our proposed algorithm GraphMOS obtained the best performance in terms of average F-measure score of 85.92 percent, which is 7.0 percent better than the second-best performing method SemanticBGS. Moreover, GraphMOS achieved the best results in seven out of eleven challenges including *Bad Weather* (94.11 percent), *Baseline* (97.10 percent), *Camera Jitter* (92.33 percent), *Night Videos* (82.11 percent), *PTZ* (85.11 percent), *Shadow* (99.01 percent), and *Thermal* (90.10 percent) while it obtained favorable performance for the remaining attributes including *Dynamic Background*, *Turbulence*, *Low Framerate*, and *Intermittent Object Motion*. In all these challenges, the sequences in *NightVideos* and *PTZ* are very challenging since the majority of the compared methods are not able to achieve more than 70.0 percent F-measure score, while GraphMOS achieved significantly high performance for these challenges.

TABLE 2
Comparison of the Qualitative Results of GraphMOS and GraphVOS Algorithms on CDNet2014, UCSD, DAVIS2016, and Youtube-VOS Datasets With Existing SOTA Methods



CDNet2014	Original	Ground Truth	SuBSENSE	PAWCS	IUTIS-5	BSUV-Net	GraphMOS (ours)
Bad Weather WetSnow in000500							
Dynamic-B Fall in002523							
PTZ Intermittent-P in001873							
Night Videos FluidHighway in000440							
UCSD	Original	Ground Truth	SuBSENSE	ROSL	DECOLOR	ViBe	GraphMOS (ours)
Birds frame_4							
Cyclists frame_9							
DAVIS2016	Original	Ground Truth	OSVOS^S	RGMP	FAVOS	OnAVOS	GraphMOS (ours)
Blackswan 00030							
Breakdance 00002							
Scooter Black 00029							
Youtube-VOS	2nd Frame	5th Frame	9th Frame	12th Frame	15th Frame	17th Frame	19th Frame
Validation 01c88b5b60							
Validation 2caa2b45c7							
Validation 3f4bacb16a							

Our algorithms perform better than the SOTA methods in these challenging scenarios.

TABLE 3
Comparisons of Average F-Measure on CDNet2014 Dataset

Challenge	SWCD	FTSG	SuBSENSE	WeSamBE	PAWCS	WisenetMD	IUTIS-5	SemanticBGS	FgSegNet v2	Mask R-CNN	BSUV-Net	GraphMOS
Bad Weather	0.8233	0.8228	0.8619	0.8608	0.8152	0.8616	0.8248	0.8260	0.3277	0.7952	**0.8713**	**0.9411**
Baseline	0.9214	0.9330	0.9503	0.9413	0.9397	0.9487	0.9567	0.9604	0.6926	0.7608	**0.9693**	**0.9710**
Camera Jitter	0.7411	0.7513	0.8152	0.7976	0.8137	0.8228	0.8332	**0.8388**	0.4266	0.7021	0.7743	**0.9233**
Dynamic-B	0.8645	0.8792	0.8177	0.7440	**0.8938**	0.8376	0.8902	**0.9489**	0.3634	0.5880	0.7967	0.8922
I-O Motion	0.7092	**0.7891**	0.6569	0.7392	0.7764	0.7264	0.7296	**0.7878**	0.2002	0.4724	0.7499	0.6455
Low-F rate	0.7374	0.6259	0.6445	0.6602	0.6588	0.6404	**0.7743**	**0.7888**	0.2482	0.5776	0.6797	0.6910
Night Videos	0.5807	0.5130	0.5599	0.5929	0.4152	0.5701	0.5290	0.5014	0.2800	0.3898	**0.6987**	**0.8211**
PTZ	0.4545	0.3241	0.3476	0.3844	0.4615	0.3367	0.4282	0.5673	0.3503	0.6214	**0.6282**	**0.8511**
Shadow	0.8779	0.8832	0.8986	0.8999	0.8913	0.8984	0.9084	**0.9478**	0.5295	0.8781	0.9233	**0.9901**
Thermal	**0.8581**	0.7768	0.8171	0.7962	0.8324	0.8152	0.8303	0.8219	0.6038	0.4986	**0.8581**	**0.9010**
Turbulence	0.7735	0.7127	0.7792	0.7737	0.6450	**0.8304**	0.7836	0.6921	0.0643	0.1707	0.7051	**0.8233**
Overall	0.7583	0.7283	0.7408	0.7446	0.7403	0.7535	0.7717	**0.7892**	0.3715	0.5868	0.7868	**0.8592**

In the I2R dataset (Table 4), GraphMOS has obtained the best performance of 91.96 percent overall, which is almost 10.0 percent better than the second-best performing BSUV-Net method. BSUV-Net+, SuBSENSE, and DECOLOR methods have obtained favorable performance, while the remaining compared methods are not able to achieve more than 60.0 percent F-measure score, which further shows the challenging nature of the sequences present in this dataset.

In the SBI2015 dataset (Table 5), the GraphMOS has also achieved the best performance in 11 out of 14 videos. Overall,

TABLE 4
Comparison of F-Measure Results Over the Sequences of I2R Dataset

Sequence	MoG	DECOLOR	ViBe	3WD	GRASTA	SuBSENSE	ROSL	ADMM	noncvxRPCA	OR1MP	BSUV-Net	BSUV-Net+	GraphMOS
Bootstrap	0.6328	0.6248	0.6134	0.5393	0.6017	0.6704	0.6449	0.5085	0.6142	0.6459	0.7133	**0.7636**	**0.9722**
Campus	0.2818	0.7652	0.4604	0.2258	0.2152	0.7498	0.1746	0.1913	0.1742	0.1714	0.8468	**0.8543**	**0.9711**
Curtain	0.6276	0.8342	0.6781	0.5127	0.7816	**0.9436**	0.7009	0.5211	0.4747	0.4624	0.9055	0.9373	**0.9899**
Escalator	0.5204	0.7183	0.5988	0.4706	0.4265	0.5756	0.3967	0.3716	0.4205	0.3673	0.6994	**0.7557**	**0.9081**
Fountain	0.7428	**0.8618**	0.5635	0.3500	0.6620	0.7937	0.2835	0.2185	0.2664	0.2646	**0.8644**	0.8306	0.8516
Hall	0.6152	0.5597	0.6377	0.5036	0.5355	0.7708	0.6751	0.3898	0.4819	0.5450	**0.8118**	0.7654	**0.8670**
Lobby	0.5760	0.5654	0.1488	0.5958	0.4059	0.2412	0.2329	0.1453	0.5157	0.2207	**0.7659**	0.7213	**0.9555**
Shopping.	0.6820	0.6800	0.5418	0.6832	0.6724	0.7594	0.6927	0.3764	0.6749	0.6198	**0.8070**	0.7619	**0.7712**
WaterSur.	0.6500	0.8873	0.8544	0.3008	0.7725	0.9365	0.6706	0.5479	0.4651	0.3607	0.9358	**0.9494**	**0.9899**
Overall	0.5920	0.7219	0.5663	0.4646	0.5635	0.7157	0.4969	0.3634	0.4542	0.4064	**0.8166**	0.8155	**0.9196**

TABLE 5
Comparison of F-Measure Results Over the Sequences of SBI2015 Dataset

Sequence	MoG	DECOLOR	ViBe	3WD	GRASTA	SuBSENSE	ROSL	ADMM	noncvxRPCA	OR1MP	BSUV-Net	BSUV-Net+	GraphMOS
Board	0.7656	0.5033	0.7377	0.3959	0.5592	0.6588	0.6440	0.1420	0.5304	0.5259	0.9560	**0.9886**	**0.9931**
CAVIAR1	0.5974	0.9183	0.8051	0.4282	0.2104	0.8783	0.8533	0.5065	0.4204	0.4653	0.9260	**0.9358**	**0.9744**
CAVIAR2	0.4597	0.4324	0.7347	0.4470	0.0463	**0.8740**	0.2966	0.2452	0.1933	0.1813	0.8572	0.8649	**0.9210**
CaVignal	0.5057	0.4827	0.3497	0.3767	0.3812	0.4080	**0.6226**	0.4305	0.4720	0.3727	0.4628	0.4773	**0.7322**
Candela	0.4927	0.5025	0.5020	0.4702	0.1929	0.6959	0.4913	0.4410	0.4730	0.6941	**0.8997**	**0.8597**	0.7551
Hall&Mon.	0.5952	0.7826	0.6017	0.4479	0.1491	0.7559	0.6081	0.1521	0.4525	0.4827	0.8931	**0.9346**	**0.9122**
HighwayI	0.6272	0.6976	0.4150	0.4123	0.5522	0.5073	0.6836	0.5170	0.5733	0.5660	**0.8440**	0.8337	**0.9880**
HighwayII	0.8144	0.8925	0.5554	0.7426	0.4833	0.8779	0.7808	0.7429	0.7335	0.7287	**0.9690**	**0.9592**	0.9547
HumanBod.	0.7553	0.8265	0.4268	0.5074	0.5352	0.8560	0.7606	0.3115	0.5765	0.5954	0.9314	**0.9503**	**0.9522**
IBMtest2	0.7108	0.8823	0.7001	0.8162	0.3825	0.9281	0.8579	0.5076	0.6714	0.6460	**0.9722**	0.9643	**0.9856**
People&Fol.	0.5920	0.2601	0.6111	0.2911	0.3460	0.4251	0.4108	0.0569	0.3924	0.3754	**0.8837**	0.6930	**0.9059**
Toscana	0.6093	0.3669	0.7307	0.3132	0.4188	0.8256	0.7083	0.2533	0.6779	0.4320	0.9110	**0.9193**	**0.9411**
Foliage	**0.5786**	0.3178	0.5539	0.3376	0.4148	0.1962	0.4341	0.2105	0.4617	0.4481	0.4371	0.3450	**0.7792**
Snellen	**0.5498**	0.4023	0.3083	0.3213	0.4104	0.2467	0.4052	0.1099	0.4345	0.4083	0.3674	0.3786	**0.7380**
Overall	0.6181	0.5906	0.5737	0.4506	0.3630	0.6524	0.6112	0.3305	0.5045	0.4944	**0.8079**	0.7932	**0.9328**

TABLE 6
Comparison of F-Measure Results Over the Videos of UCSD Background Subtraction Dataset

Sequence	MoG	DECOLOR	ViBe	3WD	GRASTA	SuBSENSE	ROSL	ADMM	noncvxRPCA	OR1MP	BSUV-Net	BSUV-Net+	GraphMOS
Birds	0.1427	0.1457	0.3354	0.1308	0.1320	**0.4832**	0.1478	0.0227	0.1432	0.1394	0.3314	0.2625	**0.7897**
Boats	0.0881	0.2179	0.1854	0.1576	0.0678	0.4550	0.1637	0.1212	0.1380	0.1100	0.4586	**0.6621**	**0.8311**
Bottle	0.1856	0.4765	0.4512	0.1364	0.1159	0.6570	0.2069	0.6589	0.1974	0.1795	**0.8528**	0.5039	**0.9011**
Chopper	0.3237	0.6214	0.4930	0.3171	0.0842	**0.6723**	0.2920	0.1250	0.3103	0.2653	0.2805	0.3020	**0.8230**
Cyclists	0.0915	0.2224	0.1211	0.1003	0.1243	0.1445	0.1366	0.1093	0.1317	0.1242	0.0051	**0.4138**	**0.7911**
Flock	0.2706	0.2943	0.2306	0.2007	0.1612	0.2492	**0.3409**	0.1088	0.3220	0.2605	0.1160	0.0025	**0.6722**
Freeway	0.2622	**0.5229**	0.4002	0.5028	0.0814	**0.5518**	0.3875	0.0816	0.3126	0.1549	0.4780	0.1185	0.4831
Hockey	0.3867	0.3449	0.4195	0.2789	0.3149	0.3611	0.4106	0.2981	0.3411	0.4296	0.6460	**0.6908**	**0.8211**
Jump	0.2679	0.3135	0.2636	0.2481	0.4175	0.2295	0.4198	0.0609	0.3180	0.3073	0.5491	**0.8697**	**0.8233**
Landing	0.0335	0.0640	0.0433	0.0457	0.0414	0.0026	0.0506	**0.0826**	0.0480	0.0442	0.0021	0.0012	**0.4122**
Ocean	0.1113	0.1315	0.1648	0.2055	0.1144	0.2533	0.1422	0.1809	0.1274	0.1252	0.4117	**0.5335**	**0.9422**
Peds	0.3731	**0.7942**	0.5257	0.7536	0.4653	0.5154	0.7418	0.6667	0.4333	0.4297	0.6958	0.6738	**0.9411**
Skiing	0.2038	**0.3473**	0.1441	0.1981	0.0927	0.2482	0.1942	0.0519	0.1812	0.1791	0.0841	0.0602	**0.7622**
Surf	0.0489	0.0647	0.0462	0.0579	0.0523	0.0467	0.0453	0.0162	0.0325	0.0317	**0.0884**	0	**0.7322**
Surfers	0.0542	0.1959	0.1189	0.0962	0.0742	0.1393	0.1184	0.1950	0.1083	0.1044	0.2612	**0.4776**	**0.7591**
Trafic	0.2188	**0.2732**	0.1445	0.2032	0.0368	0.1165	0.1042	0.1044	0.0949	0.0882	0	0	**0.6670**
Overall	0.1914	0.3144	0.2555	0.2271	0.1485	0.3203	0.2439	0.1803	0.2025	0.1858	0.3288	**0.3483**	**0.7595**

the proposed algorithm obtained a 93.28 percent F-measure score, which is approximately 12.50 percent larger than BSUV-Net+ (80.79 percent) and 13.96 percent larger than BSUV-Net (79.32 percent) methods. The sequences in this dataset show clutter background scenes and slowly moving foreground objects. Therefore, the majority of the compared methods are not able to handle the overwhelming outliers of the moving objects in these sequences efficiently as compared to GraphMOS.

Similarly, in the case of the UCSD dataset (Table 6), GraphMOS also shows the best performance of 75.95 percent. The sequences in this dataset are taken from a moving camera. Therefore, it can be noticed in the performance comparison that none of the compared methods can perform favorably better since SOTA methods such as MoG, DECOLOR, ViBe, GRASTA, SuBSENSE, and ADMM are designed to handle the static camera sequences. Our proposed GraphMOS algorithm shows better performance because of its generalization capabilities to tackle unseen videos on both static and moving cameras.

For the VOS task, our proposed algorithm GraphVOS also performs better than the SOTA methods. In the DAVIS2016 dataset (Table 7), GraphVOS achieves the best accuracy in terms of $\mathcal{J}$ and $\mathcal{F}$ measures. Overall, our proposed algorithm obtained 0.50 and 1.30 percent improvements compared to the second-best performing method STM (**+YV**). On the YouTube-VOS dataset (Table 8), GraphVOS outperforms all the SOTA methods in every measure for seen and unseen object categories. The proposed algorithm shows a 6.50 percent improvement compared to the second-best performing method PrEMVOS.

TABLE 7
Comparison of $\mathcal{J}$ and $\mathcal{F}$ Metrics on the DAVIS2016 Validation Set

Method	OL	$\mathcal{J}$ mean	$\mathcal{F}$ mean
Siam R-CNN		76.8	75.4
OSVOS^S	✓	79.8	80.6
MaskRNN	✓	80.7	80.9
FEELVOS **(+YV)**		81.1	82.2
RGMP		81.5	82.0
A-GAME **(+YV)**		82.0	82.2
FAVOS		82.4	79.5
OSVOSS	✓	85.6	86.4
OnAVOS	✓	86.1	84.9
DyeNet	✓	86.2	-
STM		84.8	88.1
STM **(+YV)**		**88.7**	**89.9**
GraphVOS		**89.2**	**91.2**

OL indicates online learning, and ***(+YV)*** *indicates the use of YouTube-VOS for training.*

TABLE 8
Comparison of $\mathcal{J}$ and $\mathcal{F}$ Metrics on the Youtube-VOS Validation Set

Method	Overall	Seen		Unseen	
		$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
OSMN	51.2	60.0	60.1	40.6	44.0
MSK	53.1	59.9	59.5	45.0	47.9
RGMP	53.8	59.5	-	45.2	-
OnAVOS	55.2	60.1	62.7	46.6	51.4
RVOS	56.8	63.6	67.2	45.5	51.0
OSVOS	58.8	59.8	60.5	54.2	60.7
S2S	64.4	71.0	70.0	55.5	61.2
A-GAME	66.1	67.8	-	**60.8**	-
PReMVOS	**66.9**	**71.4**	**75.9**	56.5	**63.7**
GraphVOS	**73.4**	**74.4**	**78.6**	**66.5**	**73.9**

TABLE 9
Performance Comparisons in Terms of Average F-Measure Score for Different Segmentation Methods Used for Graph Construction

Dataset	SuBSENSE	DeepLab	Mask R-50	Cascade RS-200	Superpixel
I2R	0.7527	0.4145	**0.8397**	0.8196	**0.8510**
CDNet2014	**0.7396**	0.6119	0.6466	0.7283	**0.8138**
SBI2015	0.6488	0.7534	0.7538	**0.7832**	**0.8611**
UCSD	0.3388	0.4088	0.5553	**0.6751**	**0.7113**
DAVIS2016	0.2243	0.7702	**0.7867**	0.7804	**0.8711**

Only handcrafted features are used to report the performance. These ablations studies involves: graph construction using DeepLab with ResNet 101 (DeepLab), Mask R-CNN with ResNet 50 (Mask R-50), Cascade Mask R-CNN with ResNeSt 200 (Cascade RS-200), SuBSENSE, and Superpixel.

TABLE 10
Performance Comparison in Terms of Average F-Measure Score of Superpixel and Block-Based Segmentation for Graph Construction Methods

Segmentation	I2R	CDNet2014	SBI2015	UCSD	DAVIS2016
Block-based	**0.8408**	**0.7494**	**0.8080**	**0.6607**	**0.8773**
Superpixel	**0.9196**	**0.8592**	**0.9328**	**0.7595**	**0.9129**

The performance is reported by using both handcrafted and deep features representation of graph nodes.

TABLE 11
Performance Comparisons in Terms of Average F-Measure Score on Five Datasets Using Distinct Node Features Representations

Features	I2R	CDNet2014	SBI2015	UCSD	DAVIS2016
Handcrafted features	0.8510	0.8210	0.8611	0.7113	0.8711
Deep Features (Conv-5)	**0.8701**	**0.8491**	**0.8722**	**0.7320**	**0.8993**
Deep Features (Conv-4)	0.8665	0.8410	0.8711	0.7222	0.8801
Hand + Deep (Conv-5)	**0.9196**	**0.8673**	**0.8952**	**0.7595**	**0.9121**

Handcrafted, deep features, and the concatenation of handcrafted and deep features (Hand + Deep (Conv-5)) are used to represent graph nodes.

TABLE 12
Performance Comparisons in Terms of Average F-Measure Score for the Number of Superpixels in the SLIC Method

Superpixels (ζ)	I2R	CDNet2014	SBI2015	UCSD	DAVIS2016
100	0.9054	0.8397	0.8632	0.7265	0.8810
200	**0.9196**	**0.8592**	0.8790	**0.7595**	0.9020
300	**0.9191**	**0.8555**	**0.8952**	**0.7496**	**0.9120**
400	0.9094	0.8473	**0.8810**	0.7406	**0.9030**
500	0.8983	0.8442	0.8734	0.7363	0.8821

5.3 Ablation Studies

Several ablation studies are performed to analyze the performances of our proposed algorithms. These ablations include the analysis of different segmentation methods used in graph construction, feature extraction, and the number of superpixels used in the SLIC method. More ablation studies are included for the CDNet2014 dataset in Appendix E in the supplementary material, available online.

Segmentation Methods for Graph Nodes. Table 9 shows the performance comparison of GraphMOS for I2R, CDNet2014, SBI2015, and UCSD datasets and GraphVOS for the DAVIS 2016 dataset using different segmentation methods for node representation during graph construction. Overall, the superpixel segmentation-based node representation for graph construction achieves the best performance as compared to other heavyweight semantic and instance segmentation methods. This is because the superpixel method segments all the homogeneous regions in the video frames, and then the graph is constructed for the semi-supervised learning task. In contrast, the instances of the videos such as moving objects, static objects, or other undesirable objects may or may not be segmented accurately by the heavyweight deep learning DeepLab and Mask R-CNN methods. Besides, the degradation in the performance of DeepLab with respect to the instance segmentation methods also suggests the unsuitability of semantic segmentation to solve the MOS problem.

Table 10 shows the performance comparisons of superpixel and block-based node representation in graph construction. The average F-measure is reported by using handcrafted plus deep features representation of the node. Superpixel-based method clearly outperforms the block-based node representation method in all datasets.

Features Analysis. To analyze the effectiveness of the proposed algorithms, we also compare the MOS and VOS results with different features extracted from graph nodes on five different datasets. Table 11 shows the performance comparisons of the proposed algorithms using different features representation. The deep features are extracted from the 4th and 5th convolutional layers of the VGG-m model [89]. Overall, the deep features show competitive performance as compared to handcrafted features, while the incorporation of both deep features (Conv-5) and handcrafted features further improves the average F-measure score in all datasets.

Number of Superpixels. Table 12 shows the performance comparison of the proposed algorithms with varying

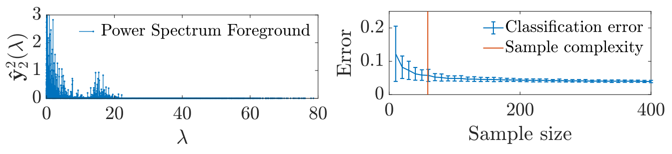


Fig. 7. Results of the experiment related to the sample complexity. Left: power spectrum of the graph signal $\hat{\mathbf{y}}_2$ related the moving objects, right: classification error versus sample size of the semi-supervised learning algorithm.

number of superpixels. The best performance for each dataset is given by 200 or 300 superpixels per image.

5.4 Sample Complexity

Fig. 7 shows the results of the experiment related to the sample complexity. The first plot in Fig. 7 shows the power spectrum of the graph signal associated with the moving objects, where the power spectrum is defined as the GFT square $\hat{\mathbf{y}}_i^2$. The second plot shows the classification error versus the sample size, and the bound of the sample complexity in the problem computed as in Eqn. (4). The sample complexity computed in this experiment is $N_s \leq 59$ with corresponding cutoff frequency $\omega = 0.0089$. The average errors are 0.0576 and 0.0393 for the sampling sizes $m = 60$ and $m = 400$, respectively, i.e., the change in the classification error is bounded by $\phi = 0.0576$ after the sample complexity. One can notice from Fig. 7 that the classification error just has a small change after a sample size of N_s.

6 Conclusion

In this study, we proposed new branch of algorithms for the tasks of MOS and VOS based on graph signal processing concepts. The pipeline of the algorithm involves segmentation, background initialization, features extraction for the representation of the nodes in a graph, construction of the graph, and finally semi-supervised learning algorithms inspired by the theory of the graph signals reconstruction. In the same way, several theoretical insights about the sample complexity and the graph signals reconstruction are explored in this work. More specifically, Corollary 2 and Theorem 3 are introduced, showing a bound for the sample complexity given a smoothness prior assumption, and two bounds for the condition number of a matrix plus a perturbation, respectively. The proposed algorithms are evaluated on six publicly available MOS and semi-supervised VOS datasets. Through an extensive series of experiments, the proposed algorithms have consistently outperformed existing state-of-the-art methods by a significant margin.

This work opens several future research directions in computer vision and machine learning. The first important direction is to further explore a generalized theory of graph signal processing in the field of MOS. The graph signals can be extended to fuzzy concepts leading to a richer representation of moving and static objects. The second direction is to explore the graph signal processing concepts applied in bounding boxes for applications such as multi-object tracking. Another important direction is to study an inductive learning framework, which aims to address the problems of real-time processing [124] for MOS and VOS. Further questions in these directions are: how can one use the structure of certain datasets to improve the generalization of SOTA deep learning methods? How can one design an algorithm to train a neural network capable of learning from the labels and the structure of a dataset? What is the relationship between the sampling of graph signals and the problems in video analysis? Perhaps, the concepts of graph signal processing, such as active semi-supervised learning and graph convolutional neural networks, could lead to new developments in the field of computer vision and end-to-end architectures for video analysis with semi-supervised learning.

Acknowledgments

This work was supported by the Khalifa University of Science and Technology under award No. RC1-2018-KUCARS. Jhony H. Giraldo, Sajid Javed, and Thierry Bouwmans equally contributed to this work.

References

[1] B. Garcia-Garcia, T. Bouwmans, and A. J. Silva, "Background subtraction in real applications: Challenges, current models and future directions," *Comput. Sci. Rev.*, vol. 35, 2020, Art. no. 100204.
[2] D. S. Lee, "Effective Gaussian mixture learning for video background subtraction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 827–832, May 2005.
[3] Y. Sheikh and M. Shah, "Bayesian modeling of dynamic scenes for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 11, pp. 1778–1792, Nov. 2005.
[4] T. S. F. Haines and T. Xiang, "Background subtraction with Dirichlet process mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 4, pp. 670–683, Apr. 2014.
[5] T. Bouwmans, "Traditional and recent approaches in background modeling for foreground detection: An overview," *Comput. Sci. Rev.*, vol. 11, pp. 31–66, 2014.
[6] T. Bouwmans *et al.*, "Decomposition into low-rank plus additive matrices for background/foreground separation: A review for a comparative evaluation with a large-scale dataset," *Comput. Sci. Rev.*, vol. 23, pp. 1–71, 2017.
[7] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "CDnet 2014: An expanded change detection benchmark dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2014, pp. 393–400.
[8] O. Oreifej, X. Li, and M. Shah, "Simultaneous video stabilization and moving object detection in turbulence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 450–462, Feb. 2013.
[9] D. S. Pham, O. Arandjelović, and S. Venkatesh, "Detection of dynamic background due to swaying movements from motion features," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 332–344, Jan. 2015.
[10] S. Li, D. Florencio, W. Li, Y. Zhao, and C. Cook, "A fusion framework for camouflaged moving foreground detection in the wavelet domain," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3918–3930, Aug. 2018.
[11] T. Bouwmans and E. H. Zahzah, "Robust PCA via principal component pursuit: A review for a comparative evaluation in video surveillance," *Comput. Vis. Image Understanding*, vol. 122, pp. 22–34, 2014.
[12] T. Bouwmans *et al.*, "Deep neural network concepts for background subtraction: A systematic review and comparative evaluation," *Neural Netw.*, vol. 117, pp. 8–66, 2019.
[13] L. Maddalena and A. Petrosino, "Towards benchmarking scene background initialization," in *Proc. Int. Conf. Image Anal. Process.*, 2015, pp. 469–476.
[14] V. Mahadevan and N. Vasconcelos, "Spatiotemporal saliency in dynamic scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 171–177, Jan. 2010.
[15] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 724–732.
[16] N. Xu *et al.*, "YouTube-VOS: Sequence-to-sequence video object segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 603–619.

[17] P. L. St-Charles, G. A. Bilodeau, and R. Bergevin, "A self-adjusting approach to change detection based on background word consensus," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2015, pp. 990–997.

[18] S. Bianco, G. Ciocca, and R. Schettini, "Combination of video change detection algorithms by genetic programming," *IEEE Trans. Evol. Comput.*, vol. 21, no. 6, pp. 914–928, Dec. 2017.

[19] O. Tezcan, P. Ishwar, and J. Konrad, "BSUV-Net: A fully-convolutional neural network for background subtraction of unseen videos," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2020, pp. 2763–2772.

[20] X. Shu, F. Porikli, and N. Ahuja, "Robust orthonormal subspace learning: Efficient recovery of corrupted low-rank matrices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3874–3881.

[21] X. Zhou, C. Yang, and W. Yu, "Moving object detection by detecting contiguous outliers in the low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 597–610, Mar. 2013.

[22] K. K. Maninis *et al.*, "Video object segmentation without temporal information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 6, pp. 1515–1530, Jun. 2019.

[23] S. Wug Oh, J.-Y. Lee, K. Sunkavalli, and S. Joo Kim, "Fast video object segmentation by reference-guided mask propagation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7376–7385.

[24] P. Voigtlaender and B. Leibe, "Online adaptation of convolutional neural networks for video object segmentation," in *Proc. Brit. Mach. Vis. Conf.*, 2017, pp. 116.1–116.13.

[25] S. W. Oh, J.-Y. Lee, N. Xu, and S. Joo Kim, "Space-time memory networks for video object segmentation with user guidance," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jul. 13, 2020, doi: 10.1109/TPAMI.2020.3008917.

[26] P. Voigtlaender, J. Luiten, P. H. S. Torr, and B. Leibe, "Siam R-CNN: Visual tracking by re-detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6577–6587.

[27] T. Bouwmans, F. El Baf, and B. Vachon, "Background modeling using mixture of Gaussians for foreground detection–A survey," *Recent Patents Comput. Sci.*, vol. 1, no. 3, pp. 219–237, 2008.

[28] L. A. Lim and H. Y. Keles, "Learning multi-scale features for foreground segmentation," *Pattern Anal. Appl.*, vol. 23, pp. 1369–1380, 2020.

[29] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2014.

[30] S. Javed, A. Mahmood, J. Dias, and N. Werghi, "Robust structural low-rank tracking," *IEEE Trans. Image Process.*, vol. 29, pp. 4390–4405, 2020.

[31] C. Li, L. Lin, W. Zuo, J. Tang, and M.-H. Yang, "Visual tracking via dynamic graph learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 11, pp. 2770–2782, Nov. 2019.

[32] S. Javed, A. Mahmood, T. Bouwmans, and S. Ki Jung, "Spatiotemporal low-rank modeling for complex scene background initialization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 6, pp. 1315–1329, Jun. 2018.

[33] S. Javed, A. Mahmood, T. Bouwmans, and S. Ki Jung, "Background-foreground modeling based on spatiotemporal sparse subspace clustering," *IEEE Trans. Image Process.*, vol. 26, no. 12, pp. 5840–5854, Dec. 2017.

[34] S. Javed, A. Mahmood, S. Al-Maadeed, T. Bouwmans, and S. Ki Jung, "Moving object detection in complex scene using spatiotemporal structured-sparse RPCA," *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 1007–1022, Feb. 2019.

[35] S. Javed *et al.*, "Cellular community detection for tissue phenotyping in colorectal cancer histology images," *Med. Image Anal.*, vol. 63, 2020, Art. no. 101696.

[36] S. Javed, A. Mahmood, N. Werghi, K. Benes, and N. Rajpoot, "Multiplex cellular communities in multi-gigapixel colorectal cancer histology images for tissue phenotyping," *IEEE Trans. Image Process.*, vol. 29, pp. 9204–9219, 2020.

[37] A. Ortega, P. Frossard, J. Kovačvić, 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.

[38] 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, May 2013.

[39] S. Chen, R. Varma, A. Sandryhaila, and J. Kovačević, "Discrete signal processing on graphs: Sampling theory," *IEEE Trans. Signal Process.*, vol. 63, no. 24, pp. 6510–6523, Dec. 2015.

[40] A. Anis, A. Gadde, and A. Ortega, "Efficient sampling set selection for bandlimited graph signals using graph spectral proxies," *IEEE Trans. Signal Process.*, vol. 64, no. 14, pp. 3775–3789, Jul. 2016.

[41] D. Romero, M. Ma, and G. B. Giannakis, "Kernel-based reconstruction of graph signals," *IEEE Trans. Signal Process.*, vol. 65, no. 3, pp. 764–778, Feb. 2017.

[42] A. Parada-Mayorga, D. L. Lau, J. H. Giraldo, and G. R. Arce, "Blue-noise sampling on graphs," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 5, no. 3, pp. 554–569, Sep. 2019.

[43] A. Anis, A. El Gamal, A. Salman Avestimehr, and A. Ortega, "A sampling theory perspective of graph-based semi-supervised learning," *IEEE Trans. Inf. Theory*, vol. 65, no. 4, pp. 2322–2342, Apr. 2019.

[44] L. Li *et al.*, "Foreground object detection from videos containing complex background," in *Proc. 11th ACM Int. Conf. Multimedia*, 2003, pp. 2–10.

[45] J. H. Giraldo and T. Bouwmans, "Semi-supervised background subtrction of unseen videos: Minimization of the total variation of graph signals," in *Proc. IEEE Int. Conf. Image Process.*, 2020, pp. 3224–3228.

[46] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013.

[47] F. R. K. Chung, *Spectral Graph Theory*. Providence, RI, USA: American Mathematical Society, 1997.

[48] X. Zhu and M. Rabbat, "Approximating signals supported on graphs," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2012, pp. 3921–3924.

[49] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Appl. Comput. Harmon. Anal.*, vol. 30, no. 2, pp. 129–150, 2011.

[50] S. K. Narang and A. Ortega, "Local two-channel critically sampled filter-banks on graphs," in *Proc. IEEE Int. Conf. Image Process.*, 2010, pp. 333–336.

[51] C. Guestrin *et al.*, "Distributed regression: An efficient framework for modeling sensor network data," in *Proc. 3rd Int. Symp. Inf. Process. Sensor Netw.*, 2004, pp. 1–10.

[52] R. Wagner, V. Delouille, and R. Baraniuk, "Distributed wavelet de-noising for sensor networks," in *Proc. 45th IEEE Conf. Decis. Control*, 2006, pp. 373–379.

[53] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[54] G. Cheung, E. Magli, Y. Tanaka, and M. K. Ng, "Graph spectral image processing," *Proc. IEEE*, vol. 106, no. 5, pp. 907–930, May 2018.

[55] I. Guskov, W. Sweldens, and P. Schröder, "Multiresolution signal processing for meshes," in *Proc. 26th Annu. Conf. Comput. Graph. Interactive Techn.*, 1999, pp. 325–334.

[56] K. Zhou, H. Bao, and J. Shi, "3D surface filtering using spherical harmonics," *Comput.-Aided Des.*, vol. 36, no. 4, pp. 363–375, 2004.

[57] R. Hammack, W. Imrich, and S. Klavžar, *Handbook of Product Graphs*. Boca Raton, FL, USA: CRC Press, 2011.

[58] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 1999, pp. 246–252.

[59] F. El Baf, T. Bouwmans, and B. Vachon, "Fuzzy integral for moving object detection," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, 2008, pp. 1729–1736.

[60] Y. Dong and G. N. DeSouza, "Adaptive learning of multi-subspace for foreground detection under illumination changes," *Comput. Vis. Image Understanding*, vol. 115, no. 1, pp. 31–49, 2011.

[61] F. Shang, J. Cheng, Y. Liu, Z.-Q. Luo, and Z. Lin, "Bilinear factor matrix norm minimization for robust PCA: Algorithms and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 9, pp. 2066–2080, Sep. 2018.

[62] N. Vaswani, T. Bouwmans, S. Javed, and P. Narayanamurthy, "Robust subspace learning: Robust PCA, robust subspace tracking, and robust subspace recovery," *IEEE Signal Process. Mag.*, vol. 35, no. 4, pp. 32–55, Jul. 2018.

[63] J. A. Ramirez-Quintana and M. I. Chacon-Murguia, "Self-adaptive SOM-CNN neural system for dynamic object detection in normal and complex scenarios," *Pattern Recognit.*, vol. 48, no. 4, pp. 1137–1149, 2015.

[64] G. T. Cinar and J. C. Príncipe, “Adaptive background estimation using an information theoretic cost for hidden state estimation,” in *Proc. Int. Joint Conf. Neural Netw.*, 2011, pp. 489–494.

[65] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[66] M. Braham and M. Van Droogenbroeck, “Deep background subtraction with scene-specific convolutional neural networks,” in *Proc. Int. Conf. Syst. Signals Image Process.*, 2016, pp. 1–4.

[67] M. Mandal, V. Dhar, A. Mishra, and S. Kumar Vipparthi, “3DFR: A swift 3D feature reductionist framework for scene independent change detection,” *IEEE Signal Process. Lett.*, vol. 26, no. 12, pp. 1882–1886, Dec. 2019.

[68] L. Yang, Y. Wang, X. Xiong, J. Yang, and A. K. Katsaggelos, “Efficient video object segmentation via network modulation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6499–6507.

[69] L. A. Lim and H. Y. Keles, “Foreground segmentation using convolutional neural networks for multiscale feature encoding,” *Pattern Recognit. Lett.*, vol. 112, pp. 256–262, 2018.

[70] Y. Wang, Z. Luo, and P. M. Jodoin, “Interactive deep learning method for segmenting moving objects,” *Pattern Recognit. Lett.*, vol. 96, pp. 66–75, 2017.

[71] J. García-González *et al.*, “Foreground detection by probabilistic modeling of the features discovered by stacked denoising autoencoders in noisy video sequences,” *Pattern Recognit. Lett.*, vol. 125, pp. 481–487, 2019.

[72] M. Sultana *et al.*, “Unsupervised deep context prediction for background estimation and foreground segmentation,” *Mach. Vis. Appl.*, vol. 30, no. 3, pp. 375–395, 2019.

[73] J. I. Jung, J. Jang, and J. Hong, “Cosine focal loss-based change detection for video surveillance systems,” in *Proc. 16th IEEE Int. Conf. Adv. Video Signal Based Surveillance*, 2019, pp. 1–6.

[74] 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.

[75] F. Yang, Q. Sun, H. Jin, and Z. Zhou, “Superpixel segmentation with fully convolutional networks,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 13961–13970.

[76] J. E. Vargas-Mu ñoz, A. S. Chowdhury, E. B. Alexandre, F. L. Galvão, P. A. Vechiatto Miranda, and A. X. Falcão, “An iterative spanning forest framework for superpixel segmentation,” *IEEE Trans. Image Process.*, vol. 28, no. 7, pp. 3477–3489, Jul. 2019.

[77] I. Pesenson, “Sampling in Paley-Wiener spaces on combinatorial graphs,” *Trans. Amer. Math. Soc.*, vol. 360, no. 10, pp. 5603–5627, 2008.

[78] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[79] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.

[80] P. L. St-Charles, G. A. Bilodeau, and R. Bergevin, “SuBSENSE: A universal change detection method with local adaptive sensitivity,” *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 359–373, Jan. 2015.

[81] Z. Cai and N. Vasconcelos, “Cascade R-CNN: High quality object detection and instance segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Nov. 28, 2019, doi: 10.1109/TPAMI.2019.2956516.

[82] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[83] H. Zhang *et al.*, “ResNeSt: Split-attention networks,” 2020, *arXiv: 2004.08955*.

[84] S. Ren *et al.*, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[85] B. D. Lucas *et al.*, “An iterative image registration technique with an application to stereo vision,” in *Proc. 7th Int. Joint Conf. Artif. Intell.*, 1981, pp. 674–679.

[86] T. Ojala, M. Pietikäinen, and T. Mäenpää, “Multiresolution grayscale and rotation invariant texture classification with local binary patterns,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[87] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, “ECO: Efficient convolution operators for tracking,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6931–6939.

[88] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, “Robust visual tracking via hierarchical convolutional features,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 11, pp. 2709–2723, Nov. 2019.

[89] K. Chatfield *et al.*, “Return of the devil in the details: Delving deep into convolutional nets,” in *Proc. Brit. Mach. Vis. Conf.*, 2014, pp. 54.1–54.12.

[90] V. N. Gangapure, S. Nanda, and A. S. Chowdhury, “Superpixel-based causal multisensor video fusion,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 6, pp. 1263–1272, Jun. 2018.

[91] M. Belkin, P. Niyogi, and V. Sindhwani, “Manifold regularization: A geometric framework for learning from labeled and unlabeled examples,” *J. Mach. Learn. Res.*, vol. 7, no. Nov, pp. 2399–2434, 2006.

[92] I. Pesenson, “Variational splines and Paley-Wiener spaces on combinatorial graphs,” *Constructive Approximation*, vol. 29, no. 1, pp. 1–21, 2009.

[93] X. Zhou and M. Belkin, “Semi-supervised learning by higher order regularization,” in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 892–900.

[94] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[95] F. Mahmood, N. Shahid, U. Skoglund, and P. Vandergheynst, “Adaptive graph-based total variation for tomographic reconstructions,” *IEEE Signal Process. Lett.*, vol. 25, no. 5, pp. 700–704, May 2018.

[96] A. Jung, A. O. Hero, III, A. C. Mara, S. Jahromi, A. Heimowitz, and Y. C. Eldar, “Semi-supervised learning in network-structured data via total variation minimization,” *IEEE Trans. Signal Process.*, vol. 67, no. 24, pp. 6256–6269, Dec. 2019.

[97] A. Chambolle and T. Pock, “A first-order primal-dual algorithm for convex problems with applications to imaging,” *J. Math. Imag. Vis.*, vol. 40, no. 1, pp. 120–145, 2011.

[98] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[99] N. Perraudin *et al.*, “UNLocBoX a Matlab convex optimization toolbox using proximal splitting methods,” 2014, *arXiv:1402.0779*.

[100] N. Komodakis and J. C. Pesquet, “Playing with duality: An overview of recent primal-dual approaches for solving large-scale optimization problems,” *IEEE Signal Process. Mag.*, vol. 32, no. 6, pp. 31–54, Nov. 2015.

[101] Y. Wu *et al.*, “Detectron2,” 2019. [Online]. Available: https://github.com/facebookresearch/detectron2

[102] N. Perraudin *et al.*, “GSPBOX: A toolbox for signal processing on graphs,” 2014, *arXiv:1408.5781*.

[103] A. Sobral and T. Bouwmans, “BGS library: A library framework for algorithm’s evaluation in foreground/background segmentation,” in *Handbook on "Background Modeling and Foreground Detection for Video Surveillance", Chapter 23*, 2014. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01227946

[104] A. Sobral, T. Bouwmans, and E. Zahzah, “LRSLibrary: Low-rank and sparse tools for background modeling and subtraction in videos,” in, *Robust Low-Rank and Sparse Matrix Decomposition: Applications in Image and Video Processing*. Boca Raton, FL, USA: CRC Press, Taylor and Francis Group, 2015.

[105] S. Isik *et al.*, “SWCD: A sliding window and self-regulated learning-based background updating method for change detection in videos,” *J. Electron. Imag.*, vol. 27, no. 2, pp. 1–11, 2018.

[106] R. Wang, F. Bunyak, G. Seetharaman, and K. Palaniappan, “Static and moving object detection using flux tensor with split gaussian models,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2014, pp. 420–424.

[107] S. Jiang and X. Lu, “WeSamBE: A weight-sample-based method for background subtraction,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 9, pp. 2105–2115, Sep. 2018.

[108] S. H. Lee *et al.*, “WisenetMD: Motion detection using dynamic background region analysis,” *Symmetry*, vol. 11, no. 5, 2019, Art. no. 621.

[109] M. Braham, S. Piérard, and M. Van Droogenbroeck, “Semantic background subtraction,” in *Proc. IEEE Int. Conf. Image Process.*, 2017, pp. 4552–4556.

[110] O. Barnich and M. Van Droogenbroeck, “ViBe: A universal background subtraction algorithm for video sequences,” *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1709–1724, Jun. 2011.

[111] J. He, L. Balzano, and A. Szlam, "Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1568–1575.

[112] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends® Optim.*, vol. 1, no. 3, pp. 127–239, 2014.

[113] Z. Kang, C. Peng, and Q. Cheng, "Robust PCA via nonconvex rank approximation," in *Proc. IEEE Int. Conf. Data Mining*, 2015, pp. 211–220.

[114] Z. Wang *et al.*, "Orthogonal rank-one matrix pursuit for low rank matrix completion," *SIAM J. Sci. Comput.*, vol. 37, no. 1, pp. A488–A514, 2015.

[115] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung, "Learning video object segmentation from static images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3491–3500.

[116] C. Ventura, M. Bellver, A. Girbau, A. Salvador, F. Marques, and X. Giro-i-Nieto, "RVOS: End-to-end recurrent network for video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5272–5281.

[117] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taix é, D. Cremers, and L. Van Gool, "One-shot video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5320–5329.

[118] J. Johnander, M. Danelljan, E. Brissman, F. S. Khan, and M. Felsberg, "A generative appearance model for end-to-end video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8945–8954.

[119] J. Luiten, P. Voigtlaender, and B. Leibe, "PReMVOS: Proposal-generation, refinement and merging for video object segmentation," in *Proc. Asian Conf. Comput. Vis.*, 2018, pp. 565–580.

[120] Y. T. Hu, J. B. Huang, and A. Schwing, "MaskRNN: Instance level video object segmentation," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 325–334.

[121] P. Voigtlaender, Y. Chai, F. Schroff, H. Adam, B. Leibe, and L. Chen, "FEELVOS: Fast end-to-end embedding learning for video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9473–9482.

[122] J. Cheng, Y. Tsai, W. Hung, S. Wang, and M. Yang, "Fast and accurate online video object segmentation via tracking parts," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7415–7424.

[123] X. Li and C. Change Loy, "Video object segmentation with joint re-identification and attention-aware mask propagation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 93–110.

[124] A. Cioppa, M. Van Droogenbroeck, and M. Braham, "Real-time semantic background subtraction," in *Proc. IEEE Int. Conf. Image Process.*, 2020, pp. 3214–3218.

Jhony H. Giraldo received the BSc and MSc degrees in electronics engineering from the University of Antioquia, Medellin, Colombia. He is currently working toward the PhD degree in computer science at the University of La Rochelle, France. His research interests include computer vision, machine learning, and graph signal processing.

Sajid Javed received the BSc degree in computer science from the University of Hertfordshire, U.K., in 2010, and the combined master's and PhD degrees in computer science from Kyungpook National University, Republic of Korea, in 2017. He was a postdoc research fellow in the University of Warwick, United Kingdom, from 2017 to 2018, where he worked on histopathological landscapes for better cancer grading and prognostication. From February 2019, he is now a research scientist in the Khalifa University Centre for Autonomous Robotics Systems (KUCARS), Abu Dhabi, UAE. His research interests include background-foreground modeling from video sequences, moving object detection from complex scenes, visual object tracking in the wild, cancer image analytics for tissue image classification, nucleus detection and classification. His research interests include developing subspace learning models (including low-rank modeling, RPCA, and matrix completion etc.), deep learning, and clustering using graph mining.

Thierry Bouwmans is an associate professor at the University of La Rochelle, France. His research interests consist mainly in the detection of moving objects in challenging environments. He has recently authored more than 30 papers in the field of back-ground modeling and foreground detection. These papers investigated particularly the use of fuzzy concepts, discriminative subspace learning models, and robust PCA. They also develop surveys on mathematical tools used in the field and particularly on robust PCA via principal component analysis. He has supervised PhD students in this field. He is the creator and administrator of the Background Subtraction Web Site. He served as the lead guest editors in two editorial works: 1) the special issue in MVA on Background Modeling for Foreground Detection in Real-World Dynamic Scenes; and 2) the Handbook on *Background Modeling and Foreground Detection for Video Surveillance* in CRC press. He has served as a reviewer for numerous international conferences and journals.