A Graph-Based Approach for Data Fusion and Segmentation of Multimodal Images

Geoffrey Iyer, Jocelyn Chanussot, *Fellow, IEEE*, and Andrea L. Bertozzi, *Member, IEEE*

***Abstract*— In the past few years, graph-based methods have proven to be a useful tool in a wide variety of energy minimization problems. In this article, we propose a graph-based algorithm for feature extraction and segmentation of multimodal images. By defining a notion of similarity that integrates information from each modality, we create a fused graph that merges the different data sources. The graph Laplacian then allows us to perform feature extraction and segmentation on the fused data set. We apply this method in a practical example, namely, the segmentation of optical and LiDAR images. The results obtained confirm the potential of the proposed method.**

***Index Terms*— Graphs, manifold learning, multimodal remote sensing, segmentation.**

I. Introduction

With the increasing availability of data, we often come upon multiple data sets, derived from different sensors that describe the same object or phenomenon. We call the sensors *modalities*, and because each modality represents some new degrees of freedom, it is generally desirable to use more modalities rather than fewer. For example, in the area of speech recognition, integrating audio data with a video of the speaker result in a much more accurate classification [1], [2]. Similarly, in medicine, it is possible to fuse the results of two different types of brain imaging to create a final image with better resolution than either of the originals [3], [4]. In this article, we also focus on multimodal images, but rather than seeking to merge our images, we instead perform feature extraction, with applications toward segmentation.

In Fig. 1, we show an example multimodal data set from the 2015 IEEE Data Fusion Challenge [5] (abbreviated as DFC2015), which consists of an optical and a LiDAR (elevation) image of a residential neighborhood in Belgium. This particular data set is interesting because of the large amount of nonredundancy between the two images. By using the LiDAR data, one can easily differentiate the roofs of the buildings from the adjacent streets even though they are roughly the same color. Conversely, the optical data allows one to separate the many different objects at the ground level even though they appear the same in the LiDAR modality. Therefore, one would expect that an algorithm that processes the two sources together would produce much more accurate segmentation results than that could be obtained by dealing with the modalities separately. We will revisit this data set in Section IV to show that this is indeed the case.

A major issue in data fusion is the difficulty of reconciling data from different modalities that, at first glance, may appear highly heterogeneous. Because of the wide variety of sensors used to acquire data, fusion methods are often tailor-made for specific problems and are not useful in general [6]. In this article, we work toward solving this problem through graph-based methods. The major advantage of using graphs lies in the ability to compare information from disparate modalities without much need for preprocessing, which makes these techniques robust to a wide variety of problems. The only requirements for implementing our graph-based multimodal method are the ability to measure the similarity between the points in the same data set, as well as a coregistration between the different sets (so the ith point in one set corresponds to the ith point in another). This situation occurs in many different image-processing problems. For example, the sets may be the images of the same scene obtained from different sensors (as is the case in our experimental data) or taken at different times.

Our method (Fig. 2) first creates a graph representation of each separate modality and then merges these representations using the coregistration assumption (see Section III-A1). From this, we obtain a single graph that constitutes a fusion of the original input information. Using this fused graph, we perform spectral clustering (see Section III-B) and semisupervised graph MBO (see Section III-C) to create a segmentation of the data. Finally, in Section IV, we show the results of the method applied to several optical/LiDAR data sets in various contexts.

The novel contributions of this article lie mainly in the method of creating the fused graph representing the full collection of input data. To the best of our knowledge, we are the first to propose a graph-based fusion method that approaches the problem at this level of generality, with any number of input data sets taken from any number of sources. In addition, this work marks the first application of the semisupervised graph MBO algorithm in the context of data fusion and

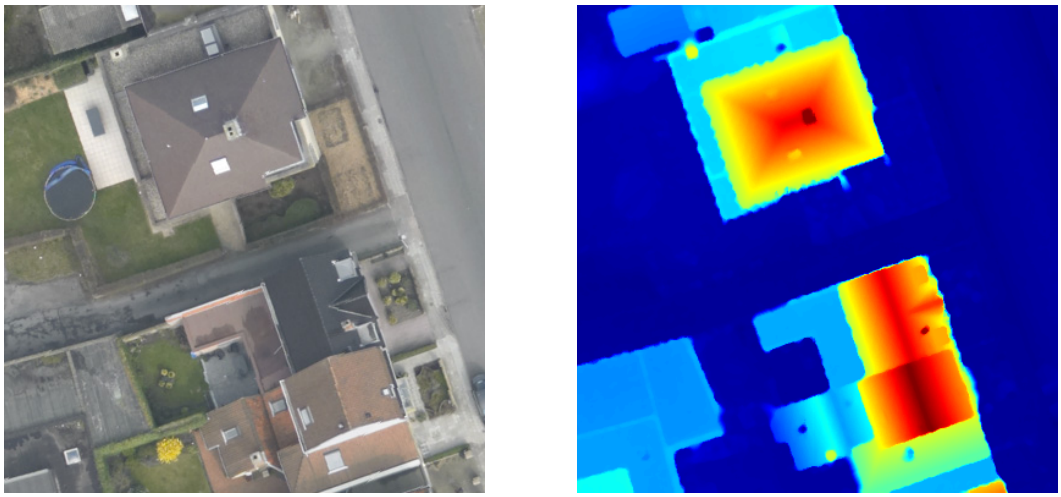
Manuscript received April 18, 2018; revised September 11, 2018 and December 12, 2018; accepted January 27, 2019. Date of publication September 25, 2020; date of current version April 22, 2021. This work was supported in part by NSF under Grant DMS-1118971, in part by the Office of Naval Research under Grant N00014-16-1-2119, in part by NSF under Grant DMS-1417674, in part by European Research Council (CHESS Project) under Grant 320684, and in part by Centre National de la Recherche Scientifique under Grant PICS-USA 263484. *(Corresponding author: Jocelyn Chanussot.)*

Geoffrey Iyer and Andrea L. Bertozzi are with the Department of Mathematics, University of California at Los Angeles, Los Angeles, CA 90095 USA (e-mail: geoff.iyer@gmail.com).

Jocelyn Chanussot is with Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab, 38000 Grenoble, France (e-mail: jocelyn.chanussot@grenoble-inp.fr).

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

Digital Object Identifier 10.1109/TGRS.2020.2971395



(a) (b)

Fig. 1. DFC2015 Input Data. (a) DFC2015 optical data. (b) DFC2015 LiDAR data.

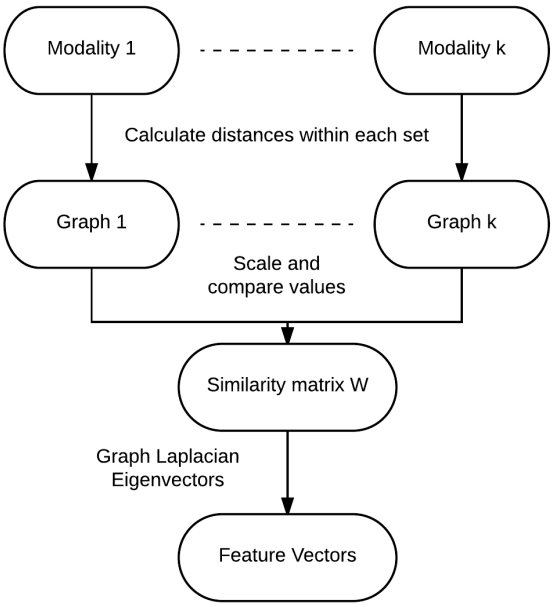


Fig. 2. Method.

remote sensing, the proper implementation of which requires nontrivial attention to the stability of the method in this new use case.

II. Related Work

One very simple algorithm for multimodal image fusion is to simply take a weighted average of the different modes. Unfortunately, this method is often too naive to produce meaningful results. In many cases, there are various objects and regions that occur in multiple images but with opposite contrast, which would cancel out in an averaged image. However, this basic idea is still worth consideration, so long as the blending step is treated with more care. Ma *et al.* [7] used structural patch decomposition to perform roughly the same task but with much better results, and Song *et al.* [8] addressed the same problem with probabilistic methods. In each of these cases, the end product is an image that contains the most relevant features from each modality. Classical segmentation algorithms can then be performed on this fused image to create the desired results. Somewhat related is the multimodal k-means method presented in [9], which minimizes a weighted average of a standard kernel k-means energy on each modality to achieve a segmentation on the data.

Another common way to fuse images is to transform each modality with some processing algorithm and then merge the data in the new feature space. Piella [10] followed this methodology, using a multiresolution (MR) transformation to process information in each modality. The benefit of this algorithm is that the transformation is fully invertible, meaning that once the data have been synthesized in the feature space, the inverse transformation can be applied to recover the fused image. Cvejic *et al.* [11] and Mitianoudis and Stathaki [12] followed the same overall strategy, using the independent component analysis (ICA) as the initial processing algorithm.

Each of the abovementioned methods first fuses the different modalities (into either a new image or into a new set of features) and then uses this fused data to create a final segmentation. However, another valid method is to instead segment each modality first and then combine the different classifications into a final result. Both [13] and [14] create a hierarchical segmentation of each modality (a chain of segmentations ranging from very coarse to very fine) and then blend these segmentations using some decision algorithm. A related field of study is segmentation combination. Given multiple segmentations of the same image (possibly obtained from different modalities), the goal is to obtain a consensus segmentation by somehow fusing the different inputs. Franek *et al.* [15] accomplished this through general ensemble clustering methods, and in [16], this is done by using probabilistic methods and random walks.

In regard to spectral graph theory, these methods have been very successfully applied to many data representation problems, with applications toward clustering and

segmentation [17]–[19]. Lu *et al.* [20] updated the low-rank representation (LRR) algorithm, including a graph spectral term to better preserve the local geometry of the input data. Liao *et al.* [21] and Debes *et al.* [22] created a sparse graph by using a k-nearest neighbors jointly over each input modality and use an RBF SVM classifier on the resulting eigenvectors to achieve a final classification. Eynard *et al.* [23] created one graph Laplacian matrix for each modality and find a single set of eigenvectors that approximately diagonalizes all Laplacians simultaneously. Zheng *et al.* [24] introduced a similar method for hyperspectral data, first performing band-selection via [25] and then solving a simultaneous diagonalization problem over Laplacians created from each band. In general, graph cuts can even be used to minimize a wide variety of energy functions [26], allowing for the use of unsupervised [27], [28] or semisupervised methods [29]. The standard theory behind this is described in [30], with a tutorial on spectral clustering given in [31]. Finally, this work was first approached by our group in the conference article [32].

III. Method

In this section, we explain the theory behind the algorithm. First, in Section III-A, we explain the graph framework used in the later segmentation steps, including the method for processing the different modalities to create objects, which can be directly compared. We then exhibit two segmentation methods that we apply to the graph object. The first, i.e., spectral clustering in Section III-B, is an unsupervised method that can be used to quickly obtain a reasonable set of "proof-of-concept" results. The second, i.e., graph MBO in Section III-C, is a semisupervised method that more carefully handles the energy minimization to obtain a stronger final result. We present a brief review of the method in algorithm 1, with the full details in the following.

Algorithm 1 Method Overview

Data: Co-registered data sets $X_1, X_2, \ldots, X_k$
Data: Number of desired classes m
Data: Semisupervised input $\hat{u}$
Result: Segmentation of $X_1, \ldots, X_k$ into m classes.
Calculate weighted graph representations $W_1, \ldots, W_k$.
Fuse to one graph W representing the full input, Section III-A1. Apply Nyström method (Section III-D) to find graph Laplacian eigenvectors. Run Spectral Clustering (Section III-B) or Graph MBO (Section III-C) using eigenvectors.

A. Graph Representation

Let k be the number of input modalities. For each $1 \leq \ell \leq k$, we have a data set, which we will label $X^\ell \subseteq \mathbb{R}^{d_\ell}$, where d_ℓ is the dimension of the data. From the coregistration assumption, we have that each set is the same size

$$n = \left|X^1\right| = \cdots = \left|X^k\right|. \tag{1}$$

Even more, they share a common indexing, which allows us to form the concatenated data set

$$X = (X^1, X^2, \ldots, X^k) \subseteq \mathbb{R}^{n \times (d_1 + \cdots + d_k)}. \tag{2}$$

We represent X using an undirected graph $G = (V, E)$. The nodes $v_i \in V$ of the graph correspond to elements of X, and we give each edge e_{ij} a weight $w_{ij} \geq 0$ representing the similarity between nodes v_i, v_j, where large weights correspond to similar nodes and small weights to dissimilar nodes. This gives rise to a symmetric similarity matrix (also called a weight matrix)

$$W = \left(w_{ij}\right)_{i,j=1}^{n}.$$

There are many different notions of "similarity" in the literature, and each has its own merits. One common similarity measure uses a radial basis function

$$w_{ij} = \exp\left(-\text{dist}\left(v_i, v_j\right)/\sigma\right) \tag{3}$$

where σ is a scaling parameter. However, this requires defining a notion of distance between two graph nodes. In this work, we create such a distance measure by considering distances between the points in each individual modality, as is explained in the following.

1) Multimodal Edge Weights: To calculate the weight matrix W, we first scale the sets $X^1, \ldots, X^k$ to make distances in each set comparable. Let $X = (X^1, \ldots, X^k) \subseteq \mathbb{R}^{n \times (d_1 + \cdots + d_k)}$ be the concatenated data set. We assume that each modality X^ℓ comes with a relevant distance function $\text{dist}_\ell(\cdot, \cdot)$ that we will use to make comparisons in that modality. Then, for $\ell = 1, \ldots, k$, define the scaling factor

$$\lambda_\ell = \text{stdev}\left(\text{dist}_\ell\left(x_i^\ell, x_j^\ell\right);\ 1 \leq i, j \leq n\right). \tag{4}$$

Now, for graph nodes $x_i, x_j \in X$, we define

$$\text{dist}\left(x_i, x_j\right) = \max\left(\frac{\text{dist}_1\left(x_i^1, x_j^1\right)}{\lambda_1}, \ldots, \frac{\text{dist}_k\left(x_i^k, x_j^k\right)}{\lambda_k}\right). \tag{5}$$

Then, define the weight matrix $W = \left(w_{ij}\right)_{1 \leq i,j \leq n}$ by

$$w_{ij} = \exp\left(-\text{dist}\left(x_i, x_j\right)\right). \tag{6}$$

Note that if each dist_ℓ on X^ℓ is a formal metric, then dist defined on X will be as well. Furthermore, if each dist_ℓ is induced by some norm on X^ℓ, then $\text{dist}(\cdot, 0)$ will be a norm on X. One benefit of this approach is that it allows us to adapt the treatment of each modality based on our knowledge of that particular source. For example, if a given modality is known to be represented poorly in the Euclidean space (the Swiss Roll data are the well-known example), we can choose a metric more suited to that particular manifold. In the context of image segmentation, we have found that using the standard Euclidean distance for each dist_ℓ produces the best results for each of our experiments in Section IV.

The purpose of choosing the maximum of input distances to combine the individual dist_ℓ is to emphasize the unique information that each data set brings. By using the maximum of all distances (i.e., the minimum of all similarities), two data points x_i, x_j are considered similar only when they are similar in every data set. For example, in Fig. 1, the gray road and the gray rooftops are considered very similar in the RGB modality but quite different in the LiDAR modality. Therefore, under this norm, the two areas will be given a low similarity score, as desired. Of course, there are many other choices

for combining the individual dist_ℓ, but through heuristics and experiments, we have found the maximum to be the most effective.

2) Graph Laplacian: Once we have created the weights, we define the normalized graph Laplacian. For each node $v_i \in V$, define the degree of the node

$$d_i = \sum_j w_{ij}. \tag{7}$$

Intuitively, the degree represents the strength of a node. Let D be the diagonal matrix with d_i as the ith diagonal entry. We then define the normalized graph Laplacian

$$L_{\text{sym}} = I - D^{-1/2} W D^{-1/2}. \tag{8}$$

For a thorough explanation of the properties of the graph Laplacian, see [30]. In this article, we will use the connection between the graph Laplacian and the graph min-cut problem, as explained in the following.

B. Spectral Clustering

To implement the first segmentation method, i.e., spectral clustering, we rephrase the data clustering problem as a graph-cut-minimization problem of the similarity matrix W. A more detailed survey of the theory can be found in [31]. Here, we state only the results necessary to implement the algorithm.

Given a partition of V into subsets $A_1, A_2, \ldots, A_m$, we define the graph N-cut

$$\text{NCut}(A_1, \ldots, A_m) = \frac{1}{2} \sum_{i=1}^{m} \frac{W(A_i, A_i^c)}{\text{vol}(A_i)} \tag{9}$$

where

$$W(A, B) = \sum_{i \in A, j \in B} w_{ij} \tag{10}$$

$$\text{vol}(A_i) = \sum_{i \in A, j \in A} w_{ij} = W(A, A). \tag{11}$$

Heuristically, minimizing the N-cut serves to minimize the connection between distinct A_i and A_j while still ensuring that each set is of a reasonable size. Without the $\text{vol}(A_i)$ term, the optimal solution often contains one large set and $m - 1$ small sets.

Solving the graph min-cut problem is equivalent to finding an $n \times m$ indicator matrix H, where

$$H_{ij} = \begin{cases} 1, & \text{if } x_i \in A_j \\ 0, & \text{else.} \end{cases} \tag{12}$$

Here, the columns of u correspond to the m different classes. Each row of H will contain a single 1, which represents the class given to that data point. It has been shown in [33] that explicitly solving this problem is an $O(|V|^{m^2})$ process. As this is infeasible in most cases, we instead introduce an approximation of the graph min-cut problem that we will solve using the graph Laplacian. The main tool here is the following algebraic fact (proven in [31]). *Fact 1:* For a given graph-cut $A_1, \ldots, A_m$, define H as earlier, and then

$$\text{NCut}(A_1, \ldots, A_m) = \text{Tr}\left(H^T L_{\text{sym}} H\right). \tag{13}$$

As explained earlier, it is infeasible to find he H that minimizes the N-Cut. Instead, we relax the problem to allow to the case of orthogonal matrices. That is, we find

$$\text{argmin}_{Y \in \mathbb{R}^{n \times m}} \text{Tr}\left(Y^T L_{\text{sym}} Y\right) \quad \text{where } Y^T Y = I. \tag{14}$$

As L_{sym} is symmetric and Y is orthogonal, this problem is solved by choosing Y to be the matrix containing the m eigenvectors of L_{sym} corresponding to the m smallest eigenvalues. Using the eigenvectors Y, we define a map $X \to \mathbb{R}^m$. For each graph node $x_i \in X$, we get a vector $y_i \in \mathbb{R}^m$ given by the ith row of Y. These y_i give the solution to the relaxed min-cut problem and, as such, can be thought of as and embedding of the original data set X into $\mathbb{R}^m$.

To obtain a solution to the original min-cut problem, we then implement some classification algorithm on the y_i. Specifically, for spectral clustering, we use k-means on the eigenvectors Y to create a final classification into m classes. Although k-means is unlikely to give an optimal classification, it is quite easy to implement, and the final results are strong enough to give a proof of concept.

Note that the eigenvectors Y found earlier are useful for many more purposes than just spectral clustering. In Section IV, we display some eigenvectors and show that they can be used to recognize objects in images. Furthermore, in Section III-C, we will use these same eigenvectors as a part of the MBO algorithm.

C. Semisupervised Graph MBO

In this section, we describe how to use eigenvectors of the graph Laplacian to segment data in a semisupervised setting. By "semisupervised," we mean that the final classification of a small number of data points (roughly 5% of all data) is used as an input to the algorithm. Following the example set in [29], [34], and [35], we formulate the problem as a minimization of the Ginzburg–Landau functional.

For the definition of the energy function, we use an $n \times m$ assignment matrix u, similar to H in (12). As before, the final output of the algorithm will be a matrix where each value is either 0 or 1, with a single 1 in each row. However, for intermediate steps of the algorithm, u will be real-valued. Heuristically, the value u_{ij} represents the strength of association between element x_i and class j. For notational convenience, we let u_i represent the ith row of u. With this notation, we define the energy function

$$\begin{aligned} E(u) = \epsilon \cdot \text{Tr}\left(u^T L_{\text{sym}} u\right) \\ + \frac{1}{\epsilon} \sum_i W(u_i) \\ + \sum_i \frac{\mu}{2} \chi(x_i) \|u_i - \hat{u}_i\|_{L_2}^2. \end{aligned} \tag{15}$$

The first term of (15) is the Dirichlet energy, similar to Section III-B. The second term is the multiwell potential

$$W(u_i) = \prod_{k=1}^{m} \frac{1}{4} \|u_i - e_k\|_{L_1}^2 \tag{16}$$

where e_k is the kth standard basis vector. These two terms together produce an approximation of the classical real

Ginzburg–Landau functional, and it has been shown in [36] that they converge to the (graph) total-variation norm

$$TV(u) = \sum_{i,j} w_{ij} \left| u_i - u_j \right| \tag{17}$$

as $\epsilon \to 0$. The last term includes the fidelity, where $\hat{u}$ represents the semisupervised input

$$\chi(x_i) = \begin{cases} 1, & \text{if } x_i \text{ is part of fidelity input} \\ 0, & \text{else} \end{cases} \tag{18}$$

and μ is a tuning parameter.

The gradient descent update associated with this energy is given by

$$\frac{\partial u}{\partial t} = -\epsilon L_{\text{sym}} u - \frac{1}{\epsilon} W'(u) - \mu \chi(x)(u - \hat{u}). \tag{19}$$

Similar to [29], [34], and [37], we propose to minimize this via an MBO algorithm. If u^n represents the nth iterate, then, to calculate u^{n+1}, we first diffuse

$$\frac{u^{n+\frac{1}{2}} - u^n}{dt} = -L_{\text{sym}} u^{n+\frac{1}{2}} - \mu(u^{n+\frac{1}{2}} - \hat{u}) + (1 - \chi(x))(u^n - \hat{u}). \tag{20}$$

Then, the threshold in each row is

$$u_i^{n+1} = e_r \quad \text{where } r = \text{argmax}_j u_{ij}^{n+\frac{1}{2}}. \tag{21}$$

This method effectively splits the energy into two parts and minimizes each alternatively. The diffusion step (20) handles the semisupervised Dirichlet energy [terms 1 and 3 in (15)], and the thresholding minimizes the potential function W [term 2 in (15)]. Note that for the diffusion equation (20), we use an implicit method to guarantee stability, as can be more clearly seen after changing coordinates in (25). The stopping criterion for this algorithm is based on the difference between two consecutive iterates u^n, u^{n+1}. In Section IV, we stop the algorithm when u^n and u^{n+1} agree on 99.99% of data points.

The diffusion calculation can be done very efficiently by using the eigendecomposition of L_{sym} (the feature vectors described in Section III-B). If we write

$$L_{\text{sym}} = H \Lambda H^T \tag{22}$$

and change coordinates

$$u^n = H a^n \tag{23}$$

$$\chi(x)(u^n - \hat{u}) = H d^n \tag{24}$$

then the diffusion step reduces to solving for coefficients

$$a_k^{n+1} = \frac{(1 + \mu dt) a_k^n - \mu dt \cdot d_k^n}{1 + \mu dt + dt \lambda_k} \tag{25}$$

where λ_k is the kth eigenvalue of L_{sym}, in the ascending order. Note that because we have $\lambda_k \geq 0$ for all k, this update step is guaranteed to be stable, regardless of the choice of parameters μ, dt.

In practice, only a small number of leading eigenvectors and eigenvalues need to be calculated in order to achieve good accuracy. Therefore, in the eigendecomposition (22), we choose a number of eigenvectors to use and truncate H to a rectangular matrix. This significantly improves the speed of the algorithm. Furthermore, in Section III-D, we discuss how to approximate the leading eigenvectors of L_{sym} without calculating the full $n \times n$ matrix.

D. Nyström Extension

Calculating the full graph Laplacian is computationally intensive, as the matrix contains n^2 entries. Instead, we use Nyström's extension to find approximate eigenvalues and eigenvectors with a heavily reduced computation time. A more complete discussion of this method can be found in [28], [29], and [38]. Here, we will present a short review of the results.

Let V denote the set of nodes of the complete weighted graph. We choose a subset $A \subset V$ of "landmark nodes" and have B its complement. Up to a permutation of nodes, we can write the weight matrix as

$$W = \begin{pmatrix} W_{AA} & W_{AB} \\ W_{BA} & W_{BB} \end{pmatrix} \tag{26}$$

where the matrix $W_{AB} = W_{BA}^T$ consists of weights between nodes in A and nodes in B, W_{AA} consists of weights between pairs of nodes in A, and W_{BB} consists of weights between pairs of nodes in B. The Nyström's extension approximates W as

$$W \approx \begin{pmatrix} W_{AA} \\ W_{BA} \end{pmatrix} W_{AA}^{-1} \begin{pmatrix} W_{AA} & W_{AB} \end{pmatrix}. \tag{27}$$

$$= \begin{pmatrix} W_{AA} & W_{BA} \\ W_{AB} & W_{BA} W_{AA}^{-1} W_{BA}^T \end{pmatrix} \tag{28}$$

where the error of approximation is determined by how well the rows of W_{AB} span the rows of W_{BB}. More precisely, if we write W as a matrix transpose times itself, $W = U^T U$, the Nyström extension estimates the unknown part of U (corresponding to W_{BB}) by orthogonally projecting it into the known part (corresponding to W_{AA} and W_{AB}) [39]. This approximation is extremely useful, as we can use it to avoid calculating W_{BB} entirely. It is in fact possible to find $|A|$ approximate eigenvectors of W using only the matrices W_{AA} and W_{AB}. This results in a significant reduction in computation time, as we compute and store matrices of size at most $|A| \times |X|$, rather than $|X| \times |X|$.

In practice, the details of choosing A are not relevant for many common use cases. As stated earlier, the quality of approximation depends on how well the landmark nodes A represent the entire space V; however, in most cases, this is easy to achieve. In most of our experiments in the following (see Section IV), we choose 100 Nyström nodes at random, and this provides sufficient accuracy to generate good results. For the DFC2018 data in Section IV-C, the situation is more complicated, as both the data set and the desired number of classes are much larger in this case. To solve this issue, we use the semisupervised data included in the MBO algorithm to ensure that our selection of eigenvectors includes nodes from every input class, as well as increase the size of A to 200. For a completely unsupervised solution to this problem, it is

also possible to choose landmark nodes by running k-means on the initial data and using the centers found.

IV. Experiment

A. Data Fusion Challenge 2015 Images

As the first test for the algorithm, recall the DFC2015 data presented in Fig. 1. This set consists of remote sensing images in both the optical and LiDAR modalities and is interesting because of the unique information brought by each source.

In Fig. 3(a) and (b), we show two example eigenvectors of the graph Laplacian. As explained in Section III-B, these vectors can be thought of as a feature of the data set, and looking at them will give us a rough idea of the final segmentation. Notice how, in Fig. 3(a), the dark gray asphalt is distinct from both the nearby grass (which is at the same elevation) and the roofs of the buildings (which are similar colors). This shows at the feature level that the algorithm is successfully using both the optical and the LiDAR data when determining what pixels can be considered similar. Based on this example vector, the classification algorithm then separates those regions in the final results. One can note the similarities between each of the example eigenvectors and the final classifications [see Fig. 3(d) and (e)].

For this image, we choose to segment the data into six classes. As the data do not come with any ground truth attached, the number 6 was chosen based purely on personal opinion. The classes given in the semisupervised term [see Fig. 3(c)] are roughly: tall buildings, mid-level buildings, asphalt (bright), asphalt (dark), white tiles, and grass. The exact choice of fidelity pixels was made by either manually choosing locations or by characteristics of the data (for example, the 1% of pixels at the highest elevation). Most importantly, these classes can all be separated using either color or LiDAR (or both).

As should be expected, the spectral clustering method [see Fig. 3(e)] does not select exactly the same six classes that we have manually identified. As this algorithm is unsupervised, there is no way of encoding our human preference into the method. Therefore, the choice of exactly how to divide the different groups of pixels is made in accordance with only the graph min-cut energy. In the end, this algorithm can still pick out the major features of the data set, but the specific decisions of exactly which classes to combine and which to separate do not agree with our human intuition. By instead using a semisupervised algorithm, such as graph MBO [see Fig. 3(d)], we can input a small amount of information (in this case, 7% of total pixels) in order to align the energy minimization with our human expectations. Therefore, the final result aligns quite well with initial expectations.

When choosing the exact parameters for the algorithm, there are two factors to consider. The choice of dt is a tradeoff between the runtime of the program and the accuracy of the final result, and the choice of μ dictates our level of confidence in the semisupervised input [see Fig. 3(c)]. For this particular example, we choose $dt = 0.1$ and $\mu = 10^3$.

For comparison, we also show the results of a more naive algorithm. In Fig. 3(f), we apply k-means directly to the concatenated (4-D) data set, without any preprocessing. As can be seen from the result, a direct application of k-means is not well suited toward handling information from disparate sources. In this particular example, the segmentation overvalues the information from the LiDAR modality and, therefore, overclassifies the buildings based on height. This, in turn, results in a poor classification of the different ground-level features, as the RGB information is not well-used.

B. Data Fusion Challenge 2013 Images

For a quantitative evaluation of our method, we apply our multimodal graph MBO algorithm to the images from the Data Fusion Challenge 2013 [22] and draw a comparison to the results of one of the winners of the contest in [21]. The data sets distributed for the contest include a hyperspectral image and a LiDAR-derived digital surface model (DSM) of the University of Houston campus and its neighboring area, both at a spatial resolution of 2.5 m. The hyperspectral image consists of 144 bands in the 380–1050-nm range [see Fig. 4(a)], and the corresponding coregistered LiDAR data represents the elevation in meters above the sea level [see Fig. 4(b)].

Along with the input data, training and validation sets were created by the Data Fusion Technical Committee, labeling 2832 pixels for the training set and 15 029 pixels for the testing set [see Fig. 4(c) and (d)]. The pixels were separated into 15 classes, as detailed in Table I. As shown, both land cover and land use classes were considered, including natural objects (e.g., grass, tree, soil, and water) and man-made objects (e.g., road, highway, and railway). Note that the "Parking Lot 1" class included parking garages at ground level and in elevated areas, and "Parking Lot 2" corresponded to parked vehicles. For each class, the size of training and validation sets was made constant (when possible) to include about 200 and 1000 samples, respectively. It is noteworthy that a large cloud shadow was present during the acquisition of the HSI; as a result, no training samples were selected in this region. However, a significant number of validation samples were collected to test the efficacy of various algorithms in dealing with cloud shadow.

We show, in Fig. 4(e), the results of our multimodal graph MBO algorithm on this data, with per-class accuracies and a comparison against [21] given in Table I. The algorithm has generally comparable results, with the exception of a piece of the highway near the right-hand side of the image. By comparing the hyperspectral and LiDAR data, one can see that this section of the road is actually a bridge over the railway below and is at a higher elevation than the other segments of the highway. Since no pixels from this bridge area are labeled in the training data, the algorithm has no knowledge that this bridge is a part of the highway and misclassifies it as a building. To prevent overfitting to this particular data set, we choose to allow this misclassification and present the results as is.

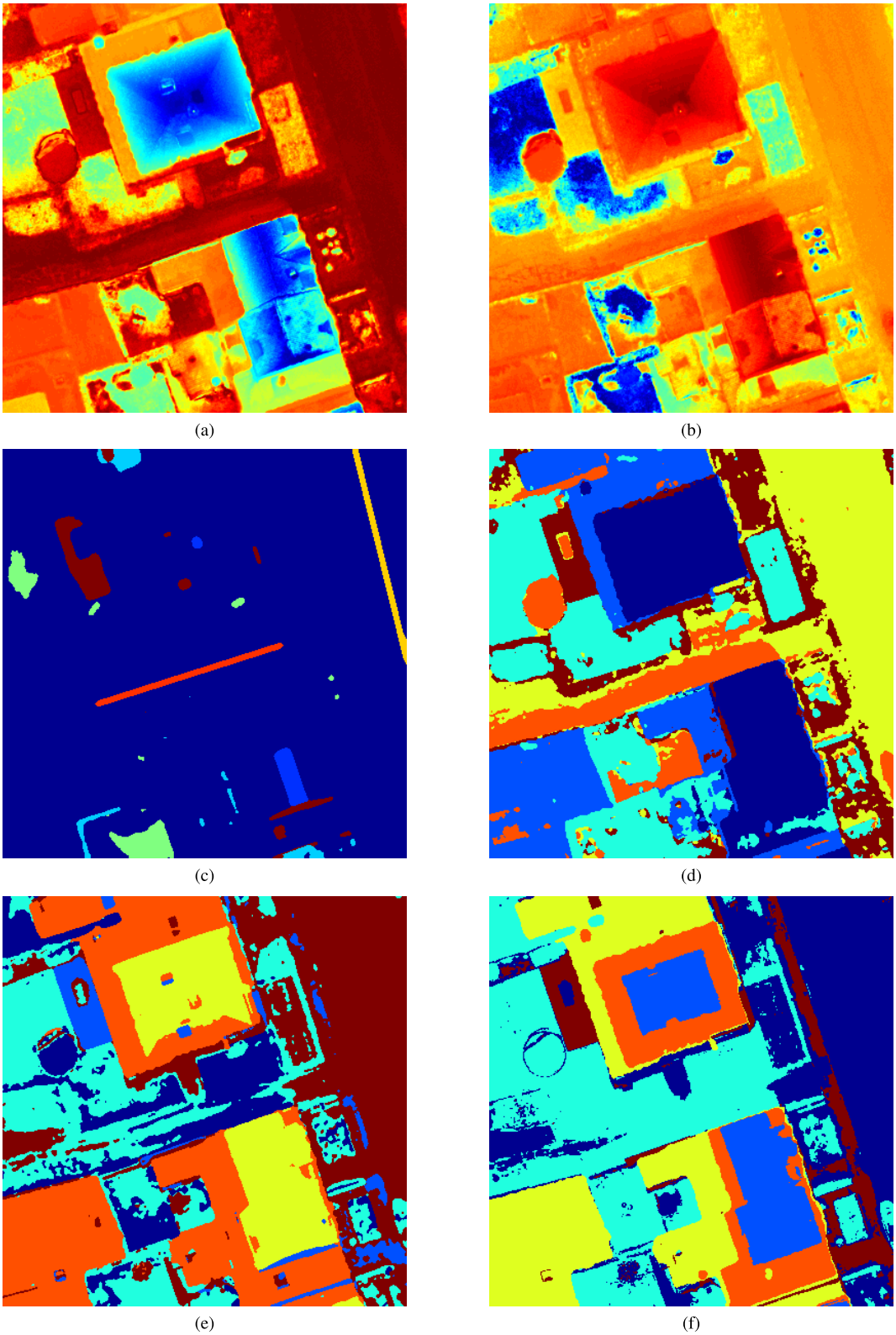


Fig. 3. DFC2015 features and segmentations. (a) Example eigenvector 1. (b) Example eigenvector 2. (c) Semisupervised Input. (d) MBO segmentation. (e) Spectral clustering segmentation. (f) Direct k-means.

C. Data Fusion Challenge 2018 Images

For a more in-depth test of the algorithm, we look at the images from the Data Fusion Challenge 2018 (DFC2018). A much more robust data set, i.e., the DFC2018, consists of 50 bands of hyperspectral data covering wavelengths 380–1050 nm, multispectral LiDAR data at three different

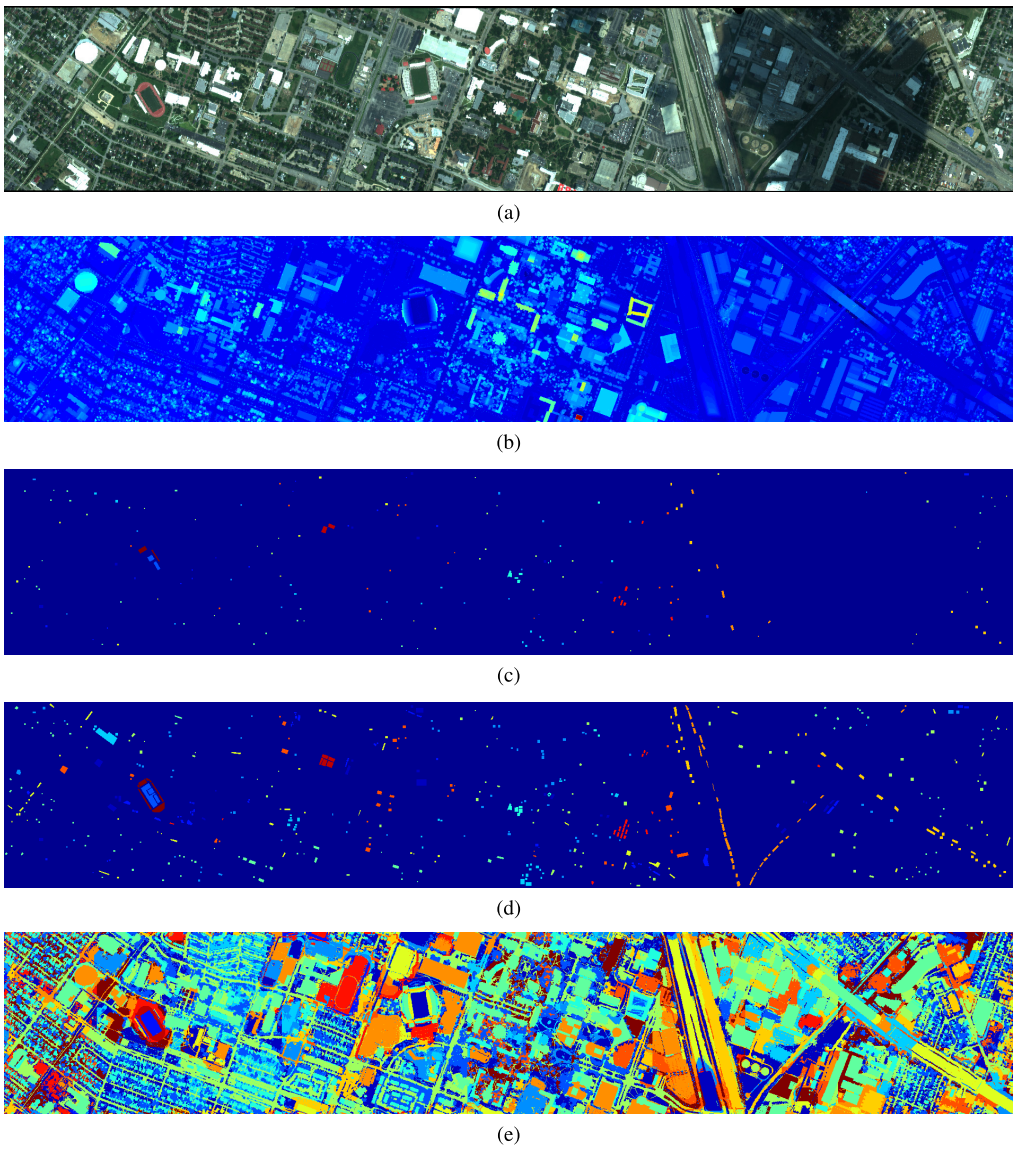


Fig. 4. DFC2013 Data. (a) Hyperspectral data (RGB bands). (b) LiDAR data. (c) Training samples. (d) Validation samples. (e) MBO segmentation.

wavelengths (1550, 1064, and 532 nm), intensity rasters for each LiDAR band, a digital surface model (DSM) of the area, and, finally, ground-truth data for roughly 30% of input pixels. The data covers an area of size 600 m $\times$ 2400 m at a resolution of 0.25^{m} per pixel, for a total of nearly 6 000 000 pixels (Fig. 5).

Featured in this data set are a wide variety of different materials. The accompanying ground truth labels 20 different classes, including different plants, water, various types of pavement, and some metal objects. We merge these 20 original classes into ten final classes (see Table II), as ground-truth data separate some objects with remarkably comparable spectral signatures (for example, roads, major thoroughfares, and highways are considered different classes in the original input). Still, among the ten final classes, we are given the opportunity to show the strength of our algorithm in considering the entire input set while differentiating between classes, as there is no one modality that fully separates all ten classes.

As explained in III-D, one difficulty in dealing with such a diverse data set is the efficacy of the Nyström eigenvectors. Recall that as a part of the MBO update step (25), we perform a coordinate change using the Graph Laplacian eigenvectors H (23). As we only calculate (an approximation of) the most influential eigenvectors, rather than the entire matrix, this coordinate change $a^n = H^T u^n$ represents a loss of overall information. Essentially, we are projecting the full data onto space spanned by the chosen eigenvectors. Therefore, it is important that each class is well represented in our choice of the Nyström landmark nodes. To accomplish this, we use the ground-truth labels given to select a few nodes from each class. We also increase the total number of Nyström eigenvectors from 100 to 200.

The application of our algorithm to the data set is also complicated by the size of the image. This particular example is more than an order of magnitude bigger than the others, and the space complexity of building the graph becomes a big issue, even when using the Nyström method

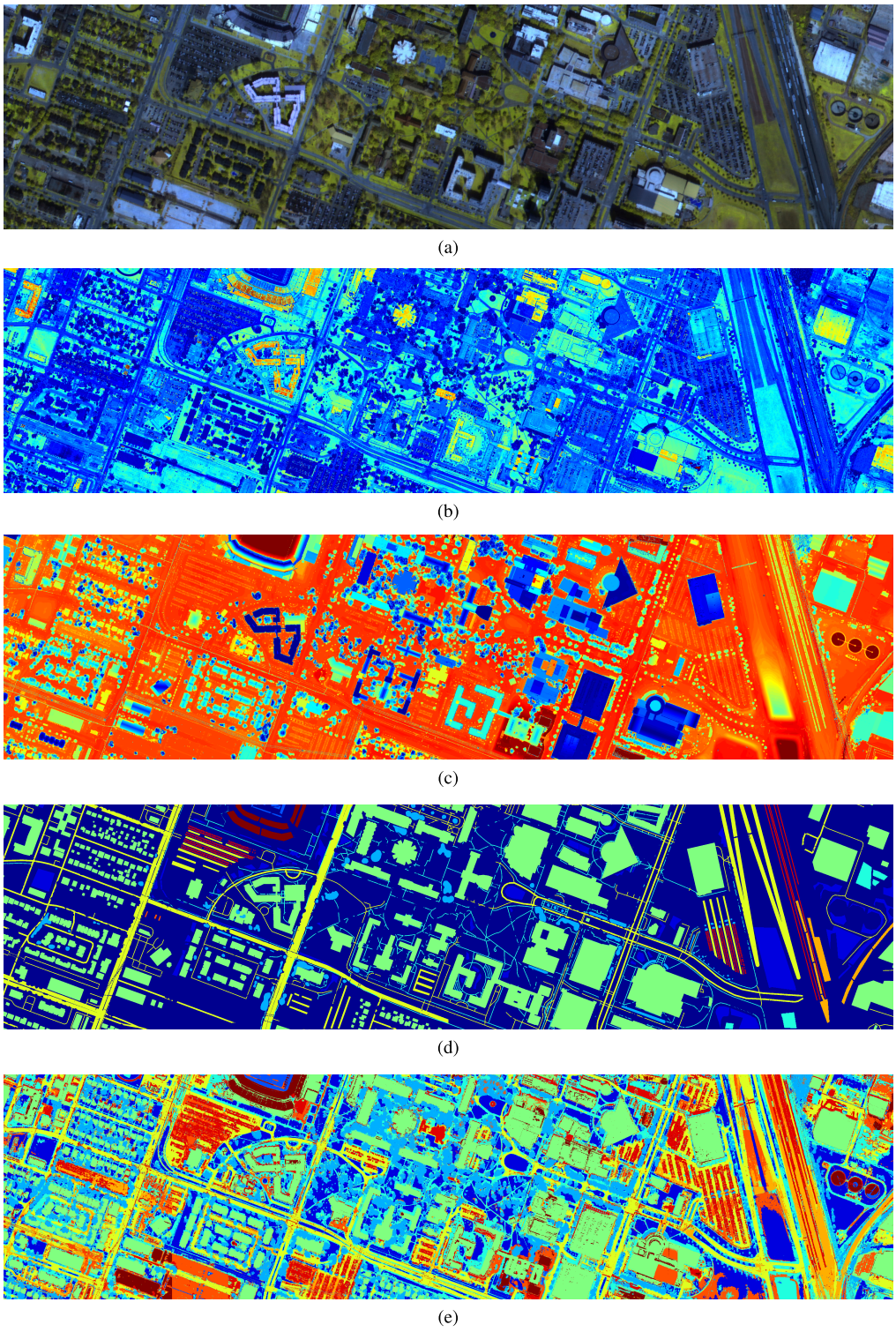


Fig. 5. DFC2018 Data. (a) Hyperspectral data (RGB bands). (b) 1064-nm intensity raster. (c) Digital surface model. (d) Ground truth. (e) MBO segmentation (using $\approx 0.1\%$ of ground truth as semisupervised input).

(see Section III-D) to reduce the matrix size. We handle the computation by running the algorithm many times over patches of size 100 000 pixels. To provide the semisupervised input for the graph MBO, we choose $\approx 0.1\%$ of the ground-truth data (with an equal distribution between class labels) and add a copy of it to each run of the algorithm. For the input parameters of the MBO, we use $dt = 0.1$ and $\mu = 10^3$. Using the ground-truth data, we can get a quantitative evaluation of the quality of our algorithm. For the pixels that are labeled in the ground truth, we correctly classify 80%.

TABLE I
RESULTS OF OUR METHOD AND OF [21] ON DFC2013 DATA

Label	Name	GGF [21]	Our method
	Healthy Grass	82.91	80.74
	Stressed Grass	99.34	87.16
	Synthetic Grass	100.00	100.00
	Trees	99.34	98.95
	Soil	100.00	99.03
	Water	95.10	98.15
	Residential	90.86	90.14
	Commercial	95.63	92.68
	Roads	89.33	74.84
	Highway	92.76	69.68
	Railway	96.58	90.45
	Parking 1	91.93	85.81
	Parking 2	74.39	81.02
	Tennis Court	100.00	99.77
	Running Track	98.73	100.00
	Overall Accuracy (%)	94.00	88.56
	Average Accuracy (%)	93.79	89.90
	κ	0.935	0.877

TABLE II
DFC2018 CLASS LABELS

Label	Substance
1	Grass
2	Artificial turf
3	Trees
4	Sidewalks and bare earth
5	Buildings
6	Roads and other paved areas
7	Railways
8	Unpaved parking lots
9	Cars and trains
10	Stadium seats

V. CONCLUSION

In conclusion, graph-based methods provide a straightforward and flexible method of combining information from multiple data sets. By considering the similarity between points in each data set, we reduce the information from each modality into something more directly comparable. This, in turn, gives us a model that is more data-driven, using the information obtained from each modality without needing to know the details about the source from which the data were captured. Therefore, the same algorithm could be applied in many different scenarios, with different types of data.

Once we have calculated and compared the different weight matrices, we can then create the graph Laplacian of the data and extract features in the form of eigenvectors. This step involves several important choices in the specific methods of comparison between modalities, and we have tuned our algorithm toward the case where each modality brings unique and relevant information about the underlying scene. These features can then be used as a part of many different data-segmentation algorithms. For this article, we use k-means on the eigenvectors as a simple proof of concept and graph MBO as a more in-depth approach. The main computational bottleneck is in the calculation of the eigenvectors, for which we have made several approximations to improve the speed of our algorithm. After this step, there are many different viable classifications in the literature.

A future area of interest is to further generalize the method by removing or weakening the coregistration assumption. This segmentation algorithm only considers cases where the two images are of the same underlying scene, where pixels correspond exactly between images. However, it would be interesting, for example, to process two images taken from different angles. In image-processing problems, coregistration is usually a reasonable assumption. However, removing this assumption would allow this algorithm to be applied to data fusion problems across a huge number of fields.

VI. ACKNOWLEDGMENT

This material is based upon research supported by the Chateaubriand Fellowship of the Office for Science and Technology of the Embassy of France in the United States.

REFERENCES

[1] G. Pomianos, C. Neti, G. Gravier, A. Garg, and A. W. Senior, "Recent advances in the automatic recognition of audiovisual speech," *Proc. IEEE*, vol. 91, no. 9, pp. 1306–1326, Sep. 2003.
[2] F. Sedighin, M. Babaie-Zadeh, B. Rivet, and C. Jutten, "Two multimodal approaches for single microphone source separation," in *Proc. 24th Eur. Signal Process. Conf. (EUSIPCO)*, Budapest, Hungary, Aug. 2016, pp. 110–114.
[3] X. Lei, P. A. Valdes-Sosa, and D. Yao, "EEG/fMRI fusion based on independent component analysis: Integration of data-driven and model-driven methods," *J. Integrative Neurosci.*, vol. 11, no. 03, pp. 313–337, Sep. 2012.
[4] S. Samadi, H. Soltanian-Zadeh, and C. Jutten, "Integrated analysis of EEG and fMRI using sparsity of spatial maps," *Brain Topogr.*, vol. 29, no. 5, pp. 661–678, Sep. 2016.
[5] M. Campos-Taberner *et al.*, "Processing of extremely high-resolution LiDAR and RGB data: Outcome of the 2015 IEEE GRSS data fusion contest—Part A: 2-D contest," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 12, pp. 5547–5559, Dec. 2016.
[6] D. Lahat, T. Adali, and C. Jutten, "Challenges in multimodal data fusion," in *Proc. 22nd Eur. Signal Process. Conf. (EUSIPCO)*, Lisbonne, Portugal, Sep. 2014, pp. 101–105.
[7] K. Ma, H. Li, H. Yong, Z. Wang, D. Meng, and L. Zhang, "Robust multi-exposure image fusion: A structural patch decomposition approach," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2519–2532, May 2017.
[8] M. Song, D. Tao, C. Chen, J. Bu, J. Luo, and C. Zhang, "Probabilistic exposure fusion," *IEEE Trans. Image Process.*, vol. 21, no. 1, pp. 341–357, Jan. 2012.
[9] S. Yu, *et al.*, "Optimized data fusion for kernel k-means clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 1031–1039, May 2012.
[10] G. Piella, "A general framework for multiresolution image fusion: From pixels to regions," *Inf. Fusion*, vol. 4, no. 4, pp. 259–280, Dec. 2003.
[11] N. Cvejic, D. Bull, and N. Canagarajah, "Region-based multimodal image fusion using ICA bases," *IEEE Sensors J.*, vol. 7, no. 5, pp. 743–751, May 2007.
[12] N. Mitianoudis and T. Stathaki, "Pixel-based and region-based image fusion schemes using ICA bases," *Inf. Fusion*, vol. 8, no. 2, pp. 131–142, Apr. 2007.
[13] G. Tochon, M. D. Mura, and J. Chanussot, "Segmentation of multimodal images based on hierarchies of partitions," in *Mathematical Morphology and Its Applications to Signal and Image Processing*. Cham, Switzerland: Springer, 2015, pp. 241–252.
[14] J. F. Randrianasoa, C. Kurtz, É. Desjardin, and N. Passat, "Multi-image segmentation: A collaborative approach based on binary partition trees," in *Mathematical Morphology and Its Applications to Signal and Image Processing*. Cham, Switzerland: Springer, 2015, pp. 253–264.

[15] L. Franek, D. D. Abdala, S. Vega-Pons, and X. Jiang, "Image segmentation fusion using general ensemble clustering methods," in *Computer Vision—ACCV*. Berlin, Germany: Springer, 2011, pp. 373–384.
[16] P. Wattuya, X. Jiang, and K. Rothaus, "Combination of multiple segmentations by a random walker approach," in *Pattern Recognition*. Berlin, Germany: Springer, 2008, pp. 214–223.
[17] T. Cour, F. Benezit, and J. Shi, "Spectral segmentation with multiscale graph decomposition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2005, pp. 1124–1131.
[18] L. Grady and E. L. Schwartz, "Isoperimetric graph partitioning for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 3, pp. 469–475, Mar. 2006.
[19] 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.
[20] X. Lu, Y. Wang, and Y. Yuan, "Graph-regularized low-rank representation for destriping of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 7, pp. 4009–4018, Jul. 2013.
[21] W. Liao, A. Pizurica, R. Bellens, S. Gautama, and W. Philips, "Generalized graph-based fusion of hyperspectral and LiDAR data using morphological features," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 3, pp. 552–556, Mar. 2015.
[22] C. Debes *et al.*, "Hyperspectral and LiDAR data fusion: Outcome of the 2013 GRSS data fusion contest," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2405–2418, Jun. 2014.
[23] D. Eynard, A. Kovnatsky, M. M. Bronstein, K. Glashoff, and A. M. Bronstein, "Multimodal manifold analysis by simultaneous diagonalization of Laplacians," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 12, pp. 2505–2517, Dec. 2015.
[24] X. Zheng, Y. Yuan, and X. Lu, "Dimensionality reduction by spatial–spectral preservation in selected bands," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 5185–5197, Sep. 2017.
[25] Y. Yuan, X. Zheng, and X. Lu, "Discovering diverse subset for unsupervised hyperspectral band selection," *IEEE Trans. Image Process.*, vol. 26, no. 1, pp. 51–64, Jan. 2017.
[26] V. Kolmogorov and R. Zabin, "What energy functions can be minimized via graph cuts?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 147–159, Feb. 2004.
[27] H. Hu, J. Sunu, and A. L. Bertozzi, "Multi-class graph Mumford–Shah model for plume detection using the MBO scheme," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*. Cham, Switzerland: Springer, 2015, pp. 209–222.
[28] J. T. Woodworth, G. O. Mohler, A. L. Bertozzi, and P. J. Brantingham, "Non-local crime density estimation incorporating housing information," *Phil. Trans. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 372, no. 2028, Nov. 2014, Art. no. 20130403.
[29] E. Merkurjev, T. Kostić, and A. L. Bertozzi, "An MBO scheme on graphs for classification and image processing," *SIAM J. Imag. Sci.*, vol. 6, no. 4, pp. 1903–1930, Jan. 2013.
[30] B. Mohar, Y. Alavi, G. Chartrand, and O. R. Oellermann, "The Laplacian spectrum of graphs," *Graph Theory, Combinatorics, Appl.*, vol. 2, pp. 871–898, Feb. 1991.
[31] U. von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, Dec. 2007.
[32] G. Iyer, J. Chanussot, and A. L. Bertozzi, "A graph-based approach for feature extraction and segmentation of multimodal images," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3320–3324.
[33] O. Goldschmidt and D. S. Hochbaum, "A polynomial algorithm for the k-cut problem for fixed k," *Math. Oper. Res.*, vol. 19, no. 1, pp. 24–37, Feb. 1994.
[34] C. Garcia-Cardona, E. Merkurjev, A. L. Bertozzi, A. Flenner, and A. G. Percus, "Multiclass data segmentation using diffuse interface methods on graphs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1600–1613, Aug. 2014.
[35] E. Merkurjev, C. Garcia-Cardona, A. L. Bertozzi, A. Flenner, and A. G. Percus, "Diffuse interface methods for multiclass segmentation of high-dimensional data," *Appl. Math. Lett.*, vol. 33, pp. 29–34, Jul. 2014.
[36] Y. Van Gennip and A. L. Bertozzi, "Γ-convergence of graph Ginzburg–Landau functionals," *Adv. Differ. Equ.*, vol. 17, nos. 11–12, pp. 1115–1180, Nov. 2012.
[37] Z. Meng, E. Merkurjev, A. Koniges, and A. L. Bertozzi, "Hyperspectral image classification using graph clustering methods," *Image Process. Line*, vol. 7, pp. 218–245, Aug. 2017.
[38] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the Nyström method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 214–225, Feb. 2004.
[39] S. Belongie, C. Fowlkes, F. Chung, and J. Malik, "Spectral partitioning with indefinite kernels using the Nyström extension," *Computer Vision—ECCV*. Berlin, Germany: Springer, 2002, pp. 531–542.