A Graph-Based Approach for Feature Extraction and Segmentation of Multimodal Images

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

I. Introduction

With the increasing availability of data we often come upon multiple datasets, 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 results in a much more accurate classification [2], [3]. 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 [4], [5] . In this paper we also focus on multimodal images, but rather than seeking to merge our images, we instead perform feature extraction, with applications toward segmentation.

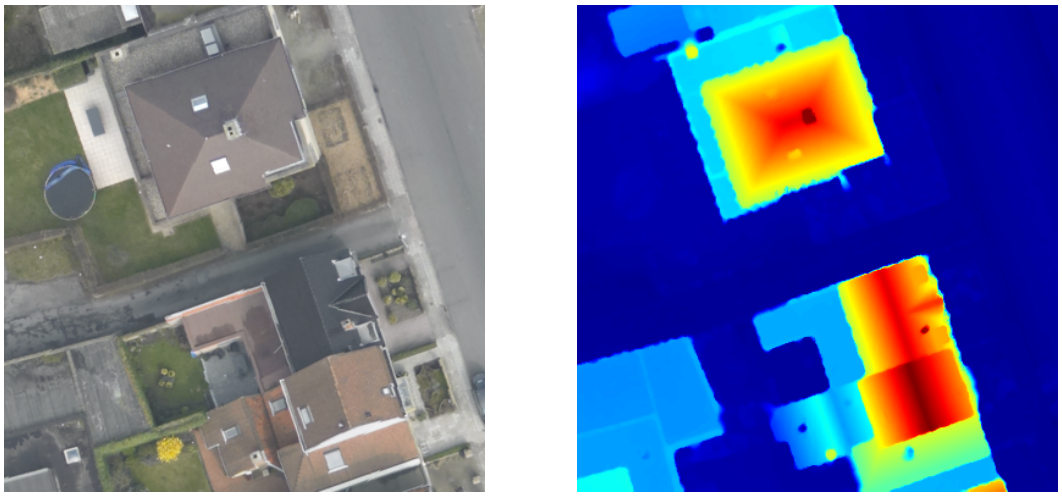
In figure 1 we show an example multimodal dataset from the 2015 IEEE Data Fusion Contest [6] (abbreviated as DFC), which consists of an optical and a lidar (elevation) image of a residential neighborhood in Belgium. This particular dataset is interesting because of the large amount of non-redundancy 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 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 could be obtained by dealing with the modalities separately. We will revisit this dataset 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 [7]. In this paper we work towards 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 pre-processing, 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 similarity between points in the same dataset, as well as a co-registration between the different sets (so the i-th point in one set corresponds to the i-th point in another). This situation occurs in many different image processing problems. For example the sets may be 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, then merges these representations using the co-registration assumption (III-A1). From this, we get a single graph that constitutes a fusion of the original input information. We then proceed to perform feature extraction and image segmentation on this graph using various well-established methods. Specifically, we extract features of the graph by finding the eigenvectors of the graph Laplacian (III-A2), then use these features as inputs to the Spectral Clustering III-B) and Graph MBO (III-C) algorithms. Finally, in IV we show the results of the method applied to several optical/lidar datasets in various different contexts.

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



(a) DFC optical data (b) DFC lidar data

Fig. 1: DFC Input Data

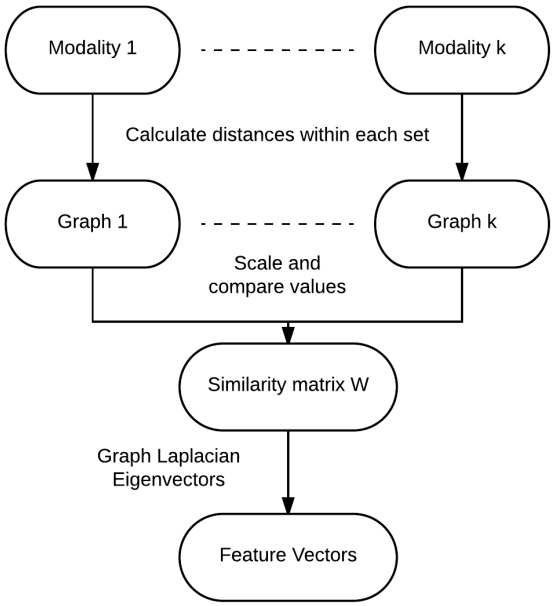


Fig. 2: The Method

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. In [8] the authors use structural patch decomposition to perform roughly the same task, but with much better results, and in [9] the authors address 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.

Another common way to fuse images is to transform each modality with some processing algorithm, then merge the data in the new feature space. In [10] the authors follow 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 has been synthesized in the feature space, the inverse transformation can be applied to recover the fused image. In [11], [12] the authors follow the same overall strategy, using Independent Component Analysis (ICA) as the initial processing algorithm.

Each of the above methods first fuses the different modalities (into either a new image, or into a new set of features), then uses this fused data to create a final segmentation. But another valid method is to instead segment each modality first, 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), 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. In [15] the authors accomplish 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 data clustering problems and image segmentation [17]–[19]. Graph-cut algorithms are quite flexible. All that is required is a well-chosen affinity function to describe the similarity between different graph nodes. Graph cuts can even be used to minimize a wide variety of energy functions [1], allowing for the use of unsupervised [20], [21] or semi-supervised methods [22]. The standard theory behind this is described in [23], with a tutorial on spectral clustering given in [24].

III. The 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, *spectral clustering* III-B, is an unsupervised method that can be used to quickly obtain a reasonable set of "proof-of-concept" results. The second, *graph MBO* III-C, is a semisupervised method that more carefully handles the energy minimization to obtain a stronger final result.

A. Graph Representation

Let k be the number of input modalities. For each $1 \leq i \leq k$, we have a dataset, which we will label $X^i \subseteq \mathbb{R}^{d_i}$, where d_i is the dimension of the data. From the co-registration assumption, we have that each set is the same size.

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

And even more, they share a common indexing, which allows us to form the concatenated dataset

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

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*)

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

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

$$w_{ij} = \exp\left(-\mathrm{dist}\,(v_i, v_j)^2 / \sigma\right), \tag{3}$$

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 points in each individual modality, as is explained below.

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 dataset. Then for $\ell = 1, \ldots, k$ define the scaling factor

$$\lambda_\ell = \mathrm{stdev}\left(\left\|x_i^\ell - x_j^\ell\right\| \;;\; 1 \leq i, j \leq n\right) \tag{4}$$

For a graph node $x \in X$, we define

$$\|x\| = \max\left(\frac{\|x^1\|}{\lambda_1}, \cdots, \frac{\|x^k\|}{\lambda_k}\right). \tag{5}$$

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

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

Note that the $\|\cdot\|$ defined above is a norm in the formal sense on the concatenated dataset X. The purpose of choosing this specific norm is to emphasize the unique information that each dataset 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 dataset. For example, in figure 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.

With any graph-based method, the choice of graph weights is always one of the leading concerns. For this reason, we have studied several different variations of equation 5, and we present our results in the appendix A.

2) The 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 i-th diagonal entry. We then define the *normalized graph Laplacian*

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

For a thorough explanation of the properties of the graph Laplacian, see [23]. In this paper, we will use the connection between the graph Laplacian and the graph min-cut problem, as explained below.

B. Spectral Clustering

To implement the first segmentation method, *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 [24]. 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*

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

Where

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

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

Heuristically, minimizing the N-cut serves to minimize the connection between distinct A_i, A_j, while still ensuring that each set is of a reasonable size. Without the $vol(A_i)$ 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 u, where

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

Here the columns of u correspond to the m different classes. Each row of u will contain a single 1, which represents the class given to that data point. It has been shown in [25] 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 fact (proven in [24]).

Fact III.1. For a given graph-cut $A_1, \ldots, A_m$, define u as above, then

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

As explained above, it is infeasible to find the u that minimizes the N-Cut. Instead we relax the problem to allow u to be an arbitrary orthogonal matrix. That is, we find

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

As L_{sym} is symmetric and u is orthogonal, this problem is solved by choosing u to be the matrix containing the m eigenvectors of L_{sym} corresponding to the m smallest eigenvalues. Using the eigenvectors u 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 i-th row of u. These y_i give the solution to the relaxed min-cut problem, and as such can be thought of as features extracted from the original dataset X.

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 u 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 u found above are useful for many more purposes than just spectral clustering. In IV we display some eigenvectors, and show they can be used to recognize objects in images. Furthermore, in III-C, we will use these same eigenvectors as 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 amount of data points (roughly 5% of all data) is used as an input to the algorithm. Following the example set in [22], [26], [27], 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 the u in (12). For intermediate steps of the algorithm, we require that

$$u_{ij} \geq 0 \quad \forall i, j \tag{15}$$

$$\sum_{j=1}^{m} u_{ij} = 1. \tag{16}$$

Heuristically, the value u_{ij} represents the probability that element x_i will be classified into class j. The final output of the algorithm will be a matrix u where each value is either 0 or 1. For notational convenience we let u_i represent the i-th row of u. With this notation, we define the energy function

$$\begin{aligned} E(u) = \epsilon \cdot \text{Tr}\left(u^T L_{sym} u\right) \\ + \frac{1}{\epsilon}\sum_i W(u_i) \\ + \sum_i \frac{\mu}{2}\lambda(x_i)\left\|u_i - \hat{u}_i\right\|_{L_2}^2. \end{aligned} \tag{17}$$

The first term of (17) is Dirichlet Energy, similar to III-B. The second term is the multiwell potential

$$W(u_i) = \prod_{k=1}^{m} \frac{1}{4}\left\|u_i - e_k\right\|_{L_1}^2, \tag{18}$$

where e_k is the k-th standard basis vector. These two terms together produce an approximation of the classical real Ginzburg-Landau functional, and it has been shown in [28] that they converge to the (graph) total-variation norm as $\epsilon \to 0$. The last term includes the fidelity, where $\hat{u}$ represents the semisupervised input,

$$\lambda(x_i) = \begin{cases} 1 & \text{if } x_i \text{ is part of fidelity input} \\ 0 & \text{else} \end{cases}, \quad (19)$$

and μ is a tuning parameter.

The gradient descent update associated to this energy is given by

$$\frac{\partial u}{\partial t} = -\epsilon L_{sym} u - \frac{1}{\epsilon} W'(u) - \mu\lambda(x)(u - \hat{u}). \quad (20)$$

Similar to [22], [26], [29], we propose to minimize this via an MBO algorithm. If u^n represents the n-th iterate, then to calculate u^{n+1} we first diffuse

$$\frac{u^{n+\frac{1}{2}} - u^n}{dt} = -L_{sym} u^n - \mu\lambda(x)(u^n - \hat{u}). \quad (21)$$

Then threshold each row

$$u_i^{n+1} = e_r \quad \text{where } r = \text{argmax}_j u_{ij}^{n+\frac{1}{2}}. \quad (22)$$

This method effectively splits the energy into two parts and minimizes each alternatively. The diffusion step (21) handles the semisupervised Dirichlet Energy (terms 1 and 3 in (17)), and the thresholding minimizes the potential function W (term 2 in (17)). 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 III-B). If we write

$$L_{sym} = X\Lambda X^T \quad (23)$$

and change coordinates

$$u^n = Xa^n \quad (24)$$

$$\mu\lambda(x)(u^n - \hat{u}) = Xd^n \quad (25)$$

then the diffusion step reduces to solving for coefficients

$$a_k^{n+1} = (1 - dt \cdot \lambda_k) \cdot a_k^n - dt \cdot d_k^n. \quad (26)$$

where λ_k is the k-th eigenvalue of L_{sym}, in ascending order.

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 23, we choose a number of eigenvectors to use, and truncate X 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. See [21], [22], [30] for a more complete discussion of this method.

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

$$W = \begin{pmatrix} W_{AA} & W_{AB} \\ W_{BA} & W_{BB} \end{pmatrix}, \quad (27)$$

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. Nyström's extension approximates W as

$$W \approx \begin{pmatrix} W_{AA} \\ W_{BA} \end{pmatrix} W_{AA}^{-1} \begin{pmatrix} W_{AA} & W_{AB} \end{pmatrix}. \quad (28)$$

where the error of approximation is determined by how well the rows of of W_{AB} span the rows of W_{BB}. As W is positive semidefinite, we can write it as a matrix transpose times itself, $W = V^T V$. In [31], the authors show that the Nyström extension estimates the unknown part of V (corresponding to W_{BB}) by orthogonally projecting it into the known part (corresponding to W_{AA}, W_{AB}). 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}, 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 will not significantly affect the final performance of the algorithm. Although it is possible to carefully choose the "landmark nodes", in many applications (including this paper) the elements of A are selected at random from the full set X. Assuming the X is not overly patterned, then it is almost guaranteed that W_{AA}, W_{AB} will be full rank. Furthermore, the amount of landmark nodes m can be chosen to be quite small without noticeably affecting performance. This makes Nyström's extension especially useful in application, as very little work is required to tune the parameters. In Section IV we use $m = 100$, and choosing a larger set A does not give a significant change in the error of approximation.

IV. Experiment

A. Data Fusion Contest 2015 Images

As a first test for the algorithm, recall the DFC data presented in figure 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 figures 3a, 3b, we show two example eigenvectors of the graph Laplacian. As explained in III-B, these vector can be thought of as feature of the dataset, and looking at them will give us a rough idea of the final segmentation. Notice how in 3a 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 a similar color). 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 3e, 3d.

For this image, we choose to segment the data into 6 classes. As the data does not come with any ground truth attached, the number 6 was chosen based purely on personal opinion. The classes given in the semisupervised term (fig 3c) 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 (ex: the 1% of pixels at 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 (fig 3e) does not select exactly the same 6 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 just the graph min cut energy. In the end, this algorithm can still pick out the major features of the dataset, but the specific decisions of exactly which classes to combine and which to separate does not agree with our human intuition. By instead using a semisupervised algorithm such as graph MBO (fig 3d), 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 several factors to consider. First, the diffusion step should be stable, which occurs when dt and μ are relatively small. Second, the final result should agree with the semisupervised input (fig 3c), which occurs when μ is relatively large. Third, the runtime should not be too long, which occurs when dt is relatively large. Balancing these different goals required multiple trial runs of the code. In this particular example we use parameters $dt = 0.1, \mu = 10^4$.

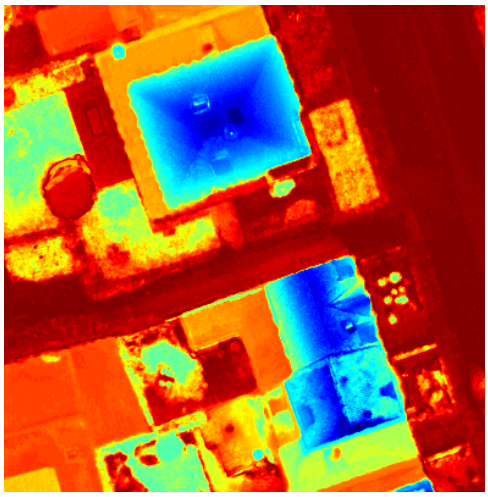
For comparison, we also show the results of a more naive algorithm. In figure 3f we apply k-means directly to the concatenated (4-dimensional) dataset, without any preprocessing. As can be seen from the result, a direct application of k-means is not well suited towards handling information from disparate 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. Umbrella Data

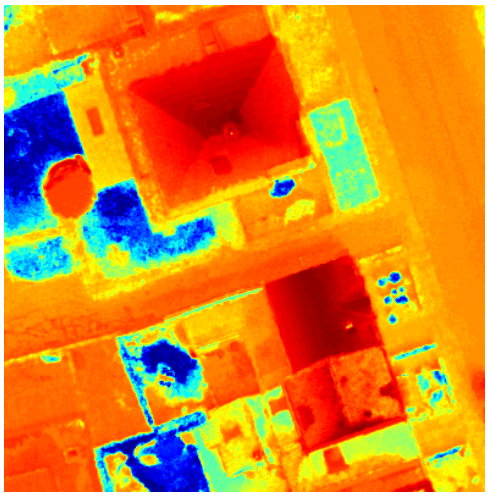
In fig 4 we show the results of the method applied to another optical/lidar set (found in [32]), which we will refer to as the umbrella data. Similar to the DFC set, the umbrella data serves as a good example because it cannot be easily analyzed using one modality alone. The umbrellas and the background walls are nearly the same shade of white, and can only be distinguished in the lidar data. Meanwhile, the different pieces of the background all lie at nearly the same depth, and can only be separated by color. As was the case with the DFC data, the final classifications 4g, 4h can be understood by looking at the individual feature vectors. In figure 4d, we see very clearly the difference the major features of the dataset: the front umbrella, the back umbrella, and the background wall. Figures 4e, 4f show more of the small details of the data, separating the many different background objects.

As was the case with the DFC data, we chose to segment this image into 6 classes based primarily on personal opinion. The classes represented in the semisupervised input are: the front umbrella, the back umbrella, the wooden cabinet in the corner, and various different colors of background objects (fig 4c). Similar to the results from the DFC dataset, we can find many major features in the spectral clustering result (fig 4g), but the exact details of the classification do not match our expectations. In particular, the foremost umbrella of this set is overclassified, which in turn forces the algorithm to combine the background objects into a small number of classes. In the graph MBO result (fig 4h), we give include the class of 5% of pixels as part of the input, and as such the classification fits the original data much more closely.

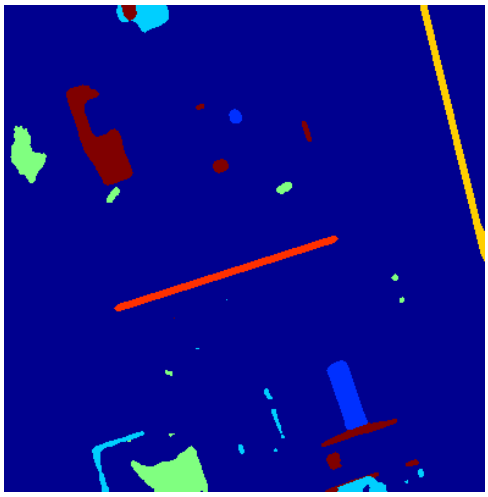
In figure 4i we again show the result of applying k-means directly to the concatenated dataset. As seen before, this naive algorithm struggles to make use of all the information present in the different modalities. In this example, the issue can be seen most clearly in the failure to separate the two umbrellas. k-means succeeds in separating many objects based on their RGB value, but



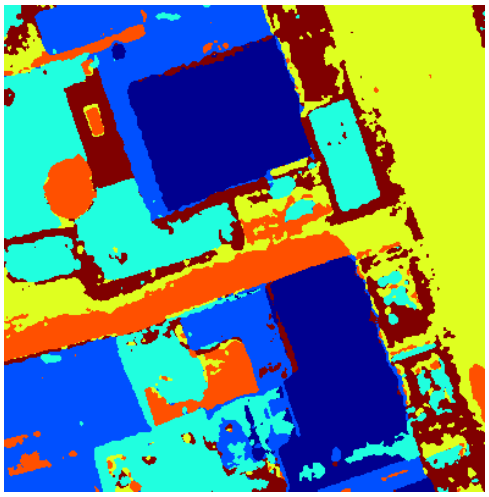
(a) Example eigenvector 1



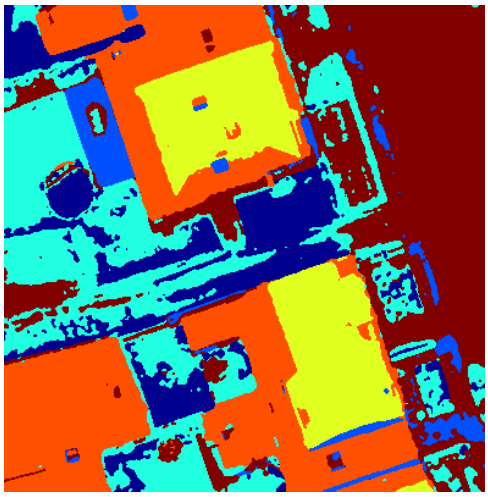
(b) Example eigenvector 2



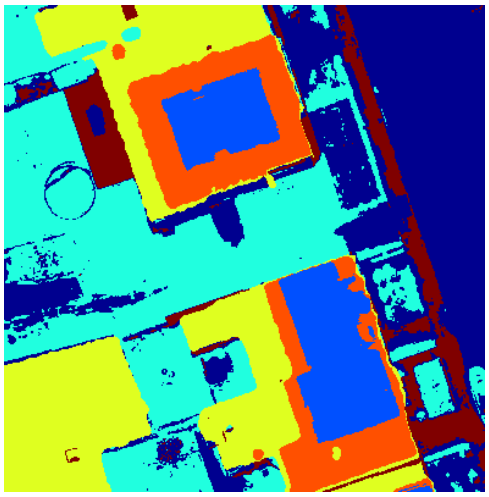
(c) Semisupervised Input



(d) MBO segmentation

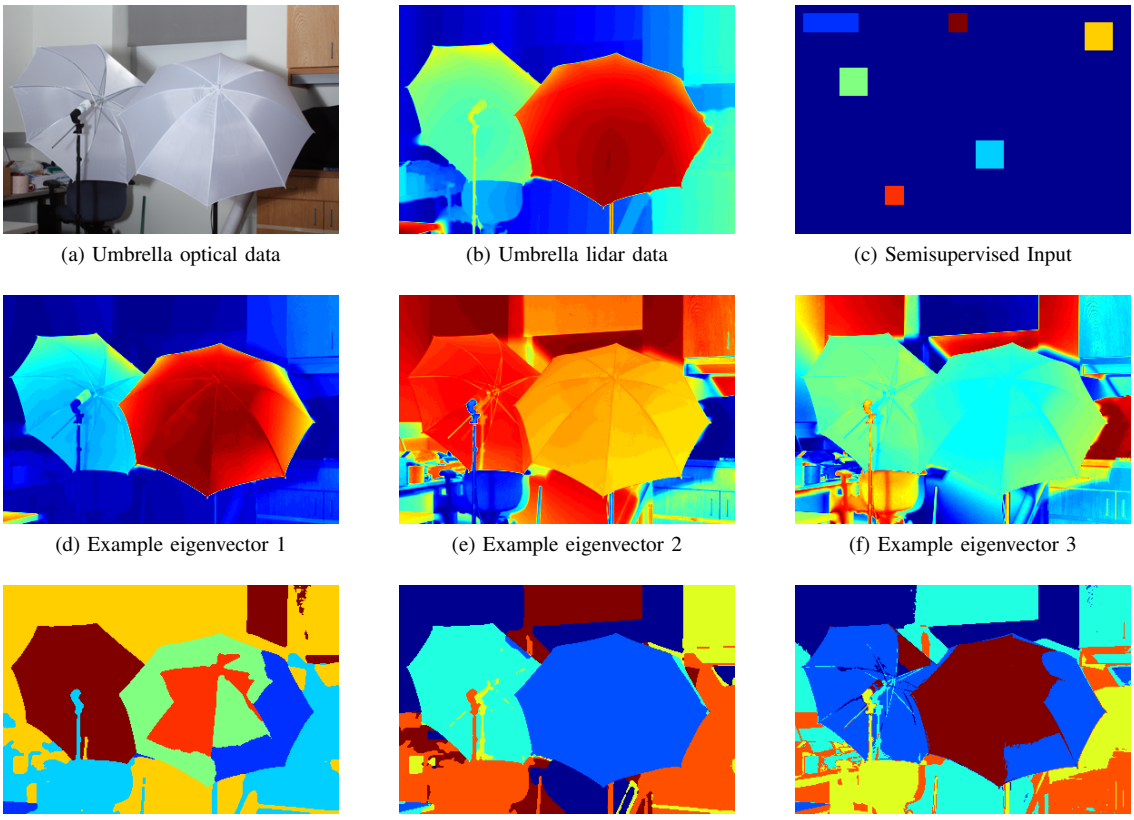


(e) Spectral clustering segmentation



(f) Direct k-means

Fig. 3: DFC features and segmentations



(a) Umbrella optical data (b) Umbrella lidar data (c) Semisupervised Input

(d) Example eigenvector 1 (e) Example eigenvector 2 (f) Example eigenvector 3

(g) Spectral clustering segmentation (h) MBO segmentation (i) Direct k-means

Fig. 4: Umbrella data results

the fact that the two umbrellas are grouped into the same class shows that the lidar information is not properly valued.

C. Jade Plant Data

Found in the same paper as the umbrella data [32], we test the method against another optical/lidar scene of a jade plant, shown in figures 5a, 5b. As with all examples shown here, there is a large amount of non-redundancy between the two input images. In particular, in this example the optical image is quite homogeneous, as it is mostly composed of shades of brown. Therefore, one would expect the addition of the lidar data to greatly aid the segmentation.

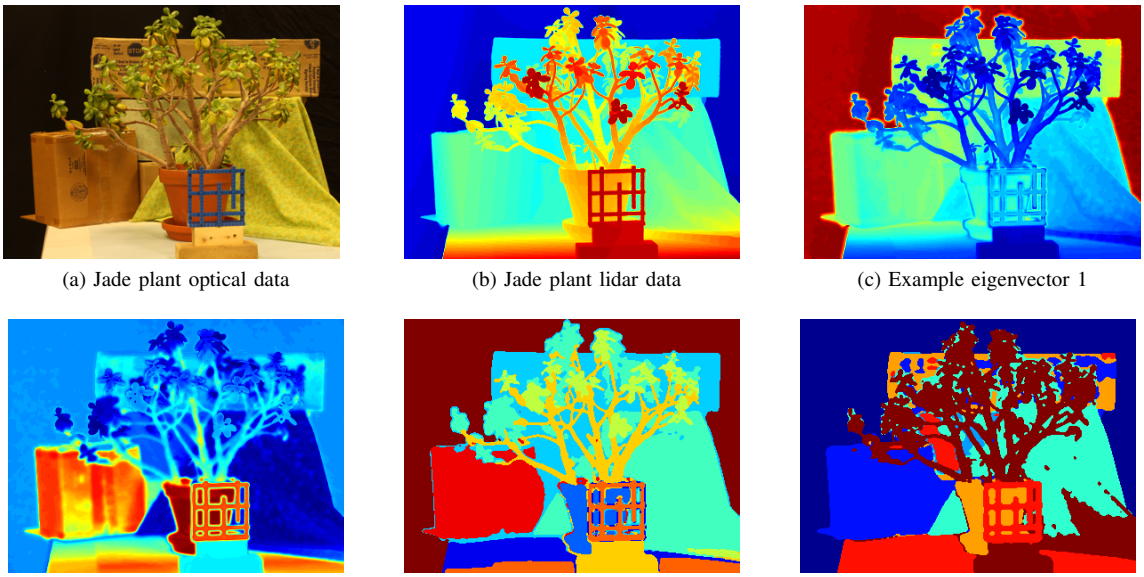
In figures 5c, 5d, we once again show a few example eigenvectors extracted from the input data. As before, we can see in each eigenvector some pieces of the final classifications 5e, 5f.

V. Conclusions

In conclusion, graph-based methods provide a straightforward and flexible method of combining information from multiple datasets. By considering the similarity between points in each individual dataset, 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 was 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. These features can then be used as part of many different data-segmentation algorithms. For this paper, 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 calculation of the eigenvectors. Once we have these, 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 co-registration assumption. This segmentation algorithm only considers cases where the two images are of the same underlying scene, where pixels correspond exactly between images.



(a) Jade plant optical data (b) Jade plant lidar data (c) Example eigenvector 1

(d) Example eigenvector 2 (e) Spectral clustering segmentation (f) MBO segmentation

Fig. 5: Jade plant data results

But it would be interesting, for example, to process two images taken from different angles. In image processing problems, co-registration 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. Acknowledgments

This work was supported by NSF grant DMS-1118971, ONR grant N00014-16-1-2119, NSF grant DMS-1417674, European Research Council (Grant no. 320684 - CHESS project), and CNRS (Grant no. PICS-USA 263484)

Appendix

In section III-A1, we create our multimodal edge weights by choosing the largest distance found throughout the different modalities. However, we could replace the $\max$ in equation 5 with a large number of different options, the easiest of which is to use a different L^p norm (thinking of $\max$ as the L^∞ norm). The question then arises, if we choose a different L^p norm, what difference should we expect in the final result? In this section we aim to answer this question, both on the level of heuristics, as well as with some more concrete observations.

The most obvious heuristic is that choosing a large p causes the edge weights to be heavily affected by individual outliers among modalities, whereas choosing a small p will provide more of an averaging effect over all the modalities. In this paper we choose to use the L^∞ norm because we expect each modality to separate some, but not all, of the objects. In other words, we believe that two pixels should be considered similar only if they are similar in every modality, and a single diffrence shown across all modalities is worth our consideration. However, for a different application - for example, when working with noisy data - a different choice of p could create a better final result.

These heuristics give an idea of the quality of difference between two choices of norms, but it is also desirable to understand the quantity of difference. If we change our choice of p norm, how much change can we expect in the graph weights, and in the resulting graph cut? Unsurprisingly, this answer is highly dependent on the number of modalities, as for $1 \leq p < q \leq \infty$ we have the inequality

$$\|x\|_p \leq \|x\|_q \leq n^{1/p-1/q} \|x\|_p , \tag{29}$$

where n is the dimension of the vector x. So when working with relatively few modalities, a different choice of p norm will not make a large difference in the distance between points (and therefore the graph weights). However, as the number of modalities increases, it is possible to have large differences in the graph weights as a result of changing p. We formalize this statement in the theorem below.

A. Theorem and Proof

We begin with some notation to simplify the statement of the theorem. Suppose we have n points $x_1, \ldots, x_n$

in some $\mathbb{R}$-space. We're interested in distances between points in different norms. So let

$$d_{ij}^p = \|x_i - x_j\|_p \tag{30}$$

$$D^p = \left\{ d_{ij}^p \,:\, 1 \le i < j \le n \right\} \tag{31}$$

In other words, for each choice of p there are $\binom{n}{2}$ values that we're interested in. More specifically, we are interested in the ordering under $\le$ for each of these sets D^p, as the graph weights are depend on the relative size of the different d_{ij}^p rather than on the absolute size. As we show below, if the ambient dimension is large enough it is possible to simultaneously control the orderings in more than one D^p by properly choosing the $x_1, \ldots, x_n$.

Theorem A.1. For any $1 \le p < q \le \infty$, it is possible to choose the $x_1, \ldots, x_n \in \mathbb{R}^n$ to simultaneously produce any arbitrary ordering under $\le$ on both D^p and D^q.

Proof. It suffices to give a proof for the case $p = 1, q = \infty$, as the L^p norm on $\mathbb{R}^n$ is a decreasing function of p, and therefore any inequalities in the case $p = 1, q = \infty$ will hold in the general case $1 \le p < q \le \infty$ as well.

To construct the $x_1, \ldots, x_n$ that produce the distances we desire, we first begin with the standard basis vectors

$$x_i = e_i \quad 1 \le i \le n \tag{32}$$

then proceed to make small edits to the x_i to achieve the desired order. Note that before making any changes, $d_{ij}^1 = 2, d_{ij}^\infty = 1$ for all i, j. To properly order the L^1 distances, we makes the following type of adjustment where necessary:

$$x_i^{new} = x_i^{old} + \epsilon \cdot e_j \quad \text{for some } \epsilon < 1, \tag{33}$$

as pictured in figure 6. This decreases d_{ij}^1 by ϵ, increases $d_{i\ell}^1$ by ϵ for $\ell \neq j$, and does not affect $d_{\ell k}^\infty$ for any ℓ, k.

Once the L^1 distances are properly set, we can fix the L^∞ distances as follows:

$$x_i^{new} = x_i^{old} + \epsilon \cdot e_j - \epsilon \cdot e_i \quad \text{for some } \epsilon < 1, \tag{34}$$

as pictured in figure 7. This decreases d_{ij}^1 by $2 \cdot \epsilon$, decreases d_{ij}^∞ by ϵ, and does not affect $d_{\ell k}^\infty$ for any ℓ, k.

Given these two possible moves, it's relatively simple to achieve the desired ordering of distances by using progressively decreasing values of epsilon. At each step one can choose epsilon sufficiently small so that the changes from the previous steps are not affected. For example, if one could use $\epsilon = 2^{-k}$ for the kth move. □

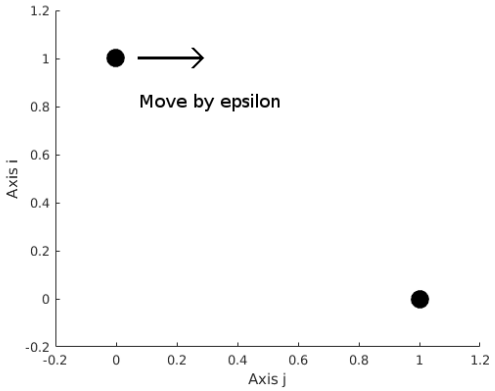


Fig. 6: Move 1

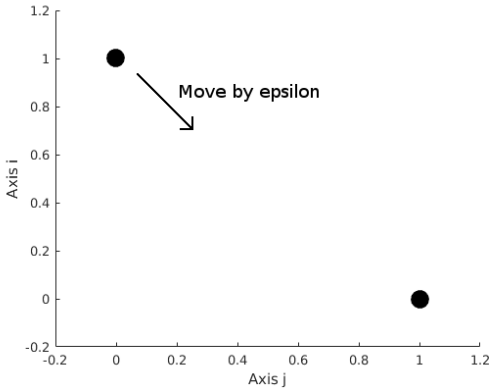


Fig. 7: Move 2

References

[1] V. Kolmogorov and R. Zabin. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, Feb 2004. II

[2] Gerasimos Potamianos, Chalapathy Neti, Guillaume Gravier, Ashutosh Garg, and Andrew W. Senior. Recent advances in the automatic recognition of audiovisual speech. *Proceedings of the IEEE*, 91(9):1306–1326, Sept 2003. I

[3] Farnaz Sedighin, Massoud Babaie-Zadeh, Bertrand Rivet, and Christian Jutten. Two Multimodal Approaches for Single Microphone Source Separation. In *24th European Signal Processing Conference (EUSIPCO 2016)*, pages 110–114, Budapest, Hungary, September 2016. I

[4] 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. Integr. Neurosci.*, 11(3):313–337, Sep 2012. I

[5] S. Samadi, H. Soltanian-Zadeh, and C. Jutten. Integrated analysis of eeg and fmri using sparsity of spatial maps. *Brain Topography*, 29(5):661–678, 2016. I

[6] M. Campos-Taberner, A. Romero-Soriano, C. Gatta, G. Camps-Valls, A. Lagrange, B. Le Saux, A. Beaupre, A. Boulch, A. Chan-Hon-Tong, S. Herbin, H. Randrianarivo, M. Ferecatu, M. Shimoni, G. Moser, and D. Tuia. Processing of extremely high-resolution lidar and rgb data: Outcome of the 2015 ieee grss data fusion contest #8211;part a: 2-d contest. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(12):5547–5559, Dec 2016. I

[7] Dana Lahat, Tülay Adali, and Christian Jutten. Challenges in Multimodal Data Fusion. In *22nd European Signal Processing Conference (EUSIPCO-2014)*, pages 101–105, Lisbonne, Portugal, September 2014. I

[8] K. Ma, H. Li, H. Yong, Z. Wang, D. Meng, and L. Zhang. Robust

multi-exposure image fusion: A structural patch decomposition approach. *IEEE Transactions on Image Processing*, PP(99):1–1, 2017. II

[9] M. Song, D. Tao, C. Chen, J. Bu, J. Luo, and C. Zhang. Probabilistic exposure fusion. *IEEE Transactions on Image Processing*, 21(1):341–357, Jan 2012. II

[10] Gemma Piella. A general framework for multiresolution image fusion: from pixels to regions. *Information Fusion*, 4(4):259 – 280, 2003. II

[11] N. Cvejic, D. Bull, and N. Canagarajah. Region-based multimodal image fusion using ica bases. *IEEE Sensors Journal*, 7(5):743–751, May 2007. II

[12] Nikolaos Mitianoudis and Tania Stathaki. Pixel-based and region-based image fusion schemes using {ICA} bases. *Information Fusion*, 8(2):131 – 142, 2007. Special Issue on Image Fusion: Advances in the State of the Art. II

[13] Guillaume Tochon, Mauro Dalla Mura, and Jocelyn Chanussot. *Segmentation of Multimodal Images Based on Hierarchies of Partitions*, pages 241–252. Springer International Publishing, Cham, 2015. II

[14] Jimmy Francky Randrianasoa, Camille Kurtz, Éric Desjardin, and Nicolas Passat. *Multi-image Segmentation: A Collaborative Approach Based on Binary Partition Trees*, pages 253–264. Springer International Publishing, Cham, 2015. II

[15] Lucas Franek, Daniel Duarte Abdala, Sandro Vega-Pons, and Xiaoyi Jiang. *Image Segmentation Fusion Using General Ensemble Clustering Methods*, pages 373–384. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. II

[16] Pakaket Wattuya, Xiaoyi Jiang, and Kai Rothaus. *Combination of Multiple Segmentations by a Random Walker Approach*, pages 214–223. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. II

[17] T. Cour, F. Benezit, and J. Shi. Spectral segmentation with multiscale graph decomposition. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 1124–1131 vol. 2, June 2005. II

[18] L. Grady and E. L. Schwartz. Isoperimetric graph partitioning for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(3):469–475, March 2006. II

[19] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, Aug 2000. II

[20] Huiyi Hu, Justin Sunu, and Andrea L. Bertozzi. *Multi-class Graph Mumford-Shah Model for Plume Detection Using the MBO scheme*, pages 209–222. Springer International Publishing, Cham, 2015. II

[21] J. T. Woodworth, G. O. Mohler, A. L. Bertozzi, and P. J. Brantingham. Non-local crime density estimation incorporating housing information. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 372(2028), 2014. II, III-D

[22] Ekaterina Merkurjev, Tijana Kostic, and Andrea L Bertozzi. An mbo scheme on graphs for classification and image processing. *SIAM Journal on Imaging Sciences*, 6:1903–1930, October 2013. II, III-C, III-C, III-D

[23] Bojan Mohar. The laplacian spectrum of graphs. *Graph Theory, Combinatorics, and Applications*, 2:871–898, 1991. II, III-A2

[24] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007. II, III-B, III-B

[25] Olivier Goldschmidt and Dorit S. Hochbaum. A polynomial algorithm for the k-cut problem for fixed k. *Mathematics of Operations Research*, 19(1):24–37, 1994. III-B

[26] C Garcia-Cardona, E Merkurjev, AL Bertozzi, A Flenner, and AG. Percus. Multiclass data segmentation using diffuse interface methods on graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1600–1613, 2014. III-C, III-C

[27] Ekaterina Merkurjev, Cristina Garcia-Cardona, Andrea L. Bertozzi, Arjuna Flenner, and Allon G. Percus. Diffuse interface methods for multiclass segmentation of high-dimensional data. *Applied Mathematics Letters*, 33:29 – 34, 2014. III-C

[28] Yves van Gennip and Andrea L. Bertozzi. γ-convergence of graph ginzburg-landau functionals. *Adv. Differential Equations*, 17(11/12):1115–1180, 11 2012. III-C

[29] Zhaoyi Meng, Ekaterina Merkurjev, Alice Koniges, and Andrea L. Bertozzi. Hyperspectral Image Classification Using Graph Clustering Methods. *Image Processing On Line*, 7:218–245, 2017. III-C

[30] Charless Fowlkes, Serge Belongie, Fan Chung, and Jitendra Malik. Spectral grouping using the nystrom method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2), February 2004. III-D

[31] Serge Belongie, Charless Fowlkes, Fan Chung, and Jitendra Malik. *Spectral Partitioning with Indefinite Kernels Using the Nyström Extension*, pages 531–542. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002. III-D

[32] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *Proceedings of the 36th German Conference on Pattern Recognition*, september 2014. IV-B, IV-C