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

Geoffrey Iyer, University of California, Los Angeles, Department of Mathematics
Jocelyn Chanussot, TODO:Figure out multiple affiliations
and Andrea L. Bertozzi, University of California, Los Angeles, Department of Mathematics

***Abstract*—In the past few years, graph-based methods have proven to be a useful tool in a wide variety of energy minimization problems [1]. 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 merge the different sources at the data level. 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.**

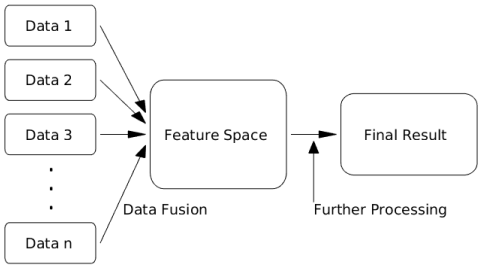


Fig. 1: Data-level fusion

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, researchers have found that integrating the audio data with a video of the speaker results in a much more accurate classification [2], [3]. Similarly, in medicine, the authors of [4] and [5] fuse the results of two different types of brain imaging to create a final image with better resolution than either of the originals. 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.

Image fusion is roughly broken into two categories depending the structure of the fusion algorithm [6]. The first of these is feature-level fusion, where each data source is collected and processed independently, then some final algorithm combines the results. These methods are generally easier to implement, as the processing step simplifies the data and allows for a more straightforward fusion process. The other category, which our method falls under, is fusion at the data level (also called pixel-level fusion when dealing with images). Here the raw data is processed as a whole, rather than as individual images creating an intermediate feature space that is informed by each dataset. This object is then used for further analysis. These terms are quite nebulous, as it is difficult to precisely define the difference between data and features, but the concept of fusion at different levels is important to consider. We believe that to optimally make use of data, the fusion step should occur as close to the data level as possible. In the general case, there should be significant information to be gained from the interactions between the different datasets, and this can be lost by prematurely processing the data. To support our belief, we compare our algorithm with several simple feature-level methods in IV, and show significantly improved results.

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. The only requirements 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 often occurs in image processing problems, where 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.

For notation, we label the sets, $X^1, X^2, \ldots, X^k$, with dimensions $d_1, d_2, \ldots, d_k$, and let

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

be the concatenated dataset. Our method extracts features from the dataset by finding eigenvectors of the graph Laplacian, then uses standard data-segmentation algorithms on these features to obtain a final classification. In section III we give the general theory behind our method, and in IV we show the results of the method applied to several optical/LIDAR dataset.

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. 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 [25], with a tutorial on spectral clustering given in [23].

III. The Method

A. Graph Laplacian

We approach this problem via graph-based methods. A more detailed survey of the theory can be found in [23]. Here we state only the results necessary to implement our algorithm.

1) The Graph Min-Cut Problem: We represent our dataset 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 *similarity matrix* (also called the *weight matrix*)

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

Since G is undirected, we require that $w_{ij} = w_{ji}$, which implies that W is a symmetric matrix. There are many different notions of "similarity" in the literature, and each has its own merits. In many applications, one defines

$$w_{ij} = -\exp\left(\|v_i - v_j\| / \sigma\right),$$

where σ is a scaling parameter. In this work we adapt this definition to apply to our multimodal dataset, as is explained in III-C.

Once the weight matrix has been defined, the data clustering problem can be rephrased as a graph-cut-minimization problem of the similarity matrix W. Given a partition of V into subsets $A_1, A_2, \ldots, A_m$, we define the *ratio graph-cut*

$$\text{RatioCut}(A_1, \ldots, A_m) = \frac{1}{2} \sum_{i=1}^{m} \frac{W(A_i, A_i^c)}{|A_i|}. \qquad (1)$$

Where

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

and the $\frac{1}{2}$ is added to account for double-counting each edge. Heuristically, minimizing the ratio 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 $|A_i|$ term, the optimal solution often contains one large set and $m-1$ small sets.

Solving the graph min-cut problem is equivalent to finding m indicator vectors $f_1, \ldots, f_m \in \mathbb{R}^n$ such that

$$f_{m,j} = \begin{cases} 1 & \text{if } x_j \in A_m \\ 0 & \text{else} \end{cases}.$$

It has been shown in [24] that explicitly solving this problem is an $O(|V|^{m^2})$ process. As this is infeasible in most cases, we instead introduce the graph Laplacian along with an approximation of the minimization problem.

2) Graph Laplacian: After forming the weight matrix W, we define the graph Laplacian. For each node $v_i \in V$, define the *degree* of the node

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

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 *graph Laplacian*

$$L = D - W. \tag{2}$$

For a thorough explanation of the properties of the graph Laplacian, see [25]. In our work we will use that L is symmetric and positive definite, as well as the following fact (proven in [23]).

Fact III.1. For a given graph-cut $A_1, \ldots, A_m$, define the $f_1, \ldots, f_m$ as above, and have $h_j = f_j / \|f_j\|$. Let H be the $n \times m$ matrix whose columns are h_j. Then $H^T H = I$, and

$$\text{RatioCut}(A_1, \ldots, A_m) = \text{Tr}\left(H^T L H\right). \tag{3}$$

As explained in III-A1, we cannot solve this problem explicitly, so instead we relax the problem to allow entries of H to take on arbitrary real values. That is, we find

$$\text{argmin}_{H\in\mathbb{R}^{n\times m}} \text{Tr}\left(H^T L H\right) \quad \text{where } H^T H = I. \tag{4}$$

As L is symmetric and H is orthogonal, this problem is solved by choosing H to be the matrix containing the m eigenvectors of L corresponding to the m smallest eigenvalues. Using the eigenvectors H 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 H. These y_i give the solution to the relaxed min-cut problem, 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 must then perform some kind of classification on the y_i to create the indicator vectors $f_1, \ldots, f_m$ as described above. There is a large variety of such methods in the literature ([20], [22] are some examples). In section IV we use k-means to segment the y_i, resulting in a well-known algorithm called *spectral clustering*. 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.

B. 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], [26] 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

$$W = \begin{pmatrix} W_{AA} & W_{AB} \\ W_{BA} & W_{BB} \end{pmatrix}, \tag{5}$$

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

$$W \approx \begin{pmatrix} W_{AA} \\ W_{BA} \end{pmatrix} W_{AA}^{-1} \begin{pmatrix} W_{AA} & W_{AB} \end{pmatrix}. \tag{6}$$

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 [27], 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_{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 choose specific "landmark

nodes", in most applications (including ours) the elements of A are selected at random from the full set X. 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 $m = n^{\frac{1}{4}}$, and choosing a larger set A does not give a significant change in the error of approximation.

C. Multimodal Edge Weights

To calculate the weight matrix W, we first scale our sets $X^1, \dots, X^k$ to make distances in each set comparable. Let $X = (X^1, \dots, X^k) \subset \mathbb{R}^{n \times (d_1 + \cdots + d_k)}$ be the concatenated dataset, and let $A \subset X$ be the collection of landmark nodes as in III-B. For simplicity of notation, rearrange the entries of X so that $A = \{x_1, \dots, x_m\}$. So $|A| = m$, and $m \ll n$. Then for $\ell = 1, \dots, k$ define the scaling factor

$$\lambda_\ell = \text{stdev}\left(\left\|x_i^\ell - x_j^\ell\right\| \ ; \ 1 \le i \le n, \ 1 \le j \le m\right) \quad (7)$$

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

$$\|x\| = \max\left(\frac{\|x^1\|}{\lambda_1}, \cdots, \frac{\|x^k\|}{\lambda_k}\right). \quad (8)$$

Then define the weight matrix W (using the Nyström Extension), by

$$W = \begin{pmatrix} W_{AA} \\ W_{AB} \end{pmatrix} = (w_{ij})_{1 \le i \le n, 1 \le j \le m} \quad (9)$$

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

Note that the $\|\cdot\|$ defined above is a norm on the concatenated dataset X. We specifically choose to use the maximum of the individual measurements to emphasize the unique information that each dataset brings. With this norm, two data points x_i, x_j are considered similar only when they are similar in each dataset.

IV. Experiment

A. Data Fusion Contest 2015 Images

We test our algorithm on an optical/LIDAR dataset from the 2015 IEEE Data Fusion Contest [28] (abbreviated as DFC), shown in figure 2a, 2b. The data consists of an RGB image and an elevation map of a residential neighborhood in Belgium. We choose this particular scene because of the large amount of non-redundancy between the two images. The lidar data is effective at differentiating the roofs of the buildings from the adjacent streets, and the optical data is useful for segmenting the many different objects at ground-level. In figures 2c, 2d we show the results of spectral clustering performed using each modality separately. The issues with single-modality segmentation can be seen immediately, as both segmentations miss out on key features of the data.

In figure 3a, 3b we show the results of our method. 3a is one example eigenvector of the graph Laplacian. As explained in III-A2, this vector can be considered one feature of our dataset, and approximates a segmentation of the image into 2 groups. Notice how in this eigenvector the dark-grey street is highlighted, while both the light-grey sidewalk (which is at the same elevation) and the nearby roof (which is the same color) are dark. This shows at the feature level that our algorithm is successfully using both the optical and the lidar data when determining what pixels can be considered similar. The difference shown in this example vector then causes the classification algorithm to separate those regions in the final result 3b. This last figure was obtained using a total of (TODO: Update this number) 12 eigenvectors (not pictured here), grouped into 5 classes.

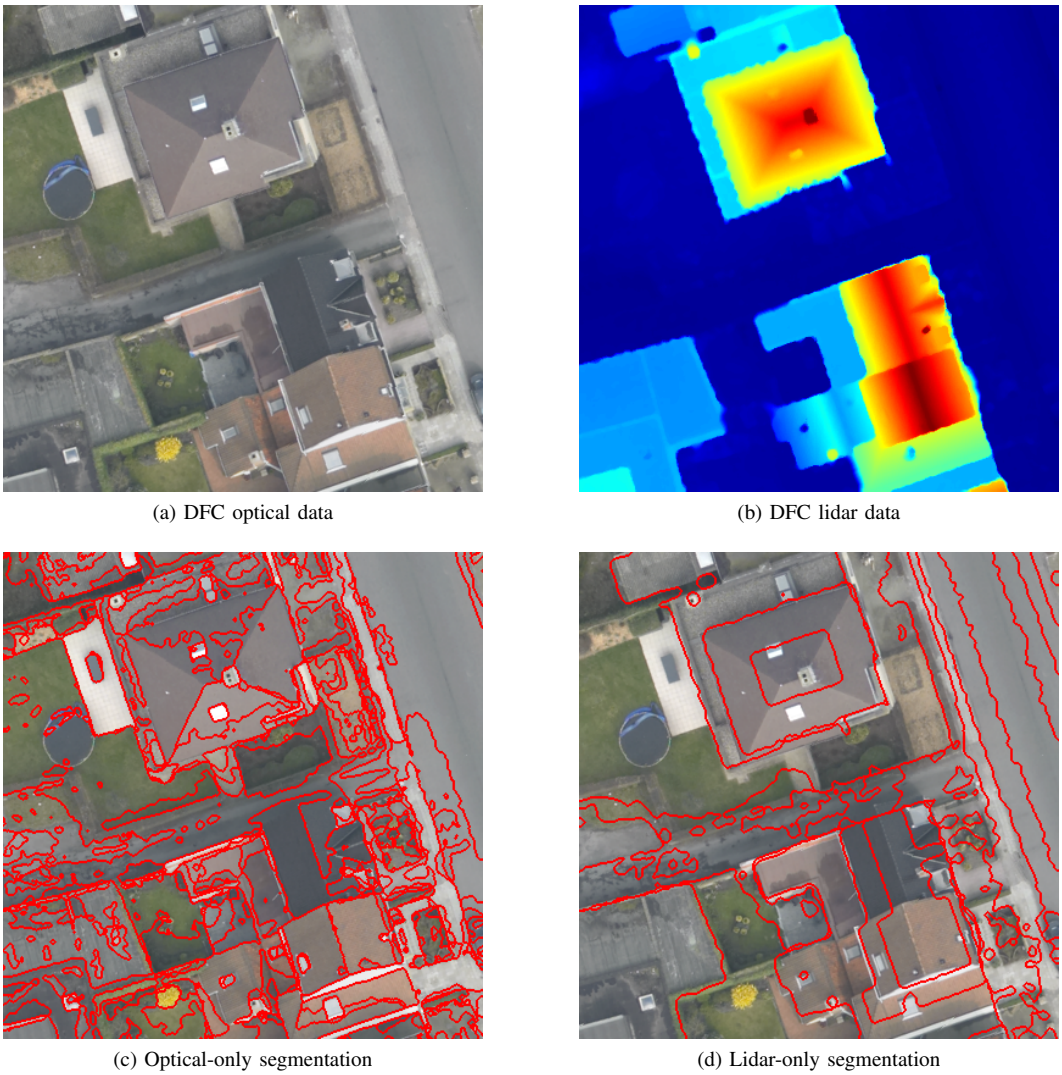
For comparison, we show the results of two other classification methods in figures 3c and 3d. The product method in 3c is a basic decision-level data fusion method. Here we use similar graph methods to segment each individual modality, then take the product of these classifications to create the final segmentation (in the product, two pixels lie in the same class iff they agree in both of the individual segmentations). This method is generally slower, as it requires running the segmentation process twice, and the results range from slightly worse to much worse than our algorithm, depending on the data. In figure 3d we apply kmeans to the original dataset.

To give some quantitative comparisons, we must first define an error metric. Unfortunately, for a given segmentation of an image computing the graph-cut error as described in III-A1 is an $O(n^2)$ calculation, and requires the full weight matrix W. To avoid this, we instead measure the error of segmentation by how the data $X = (X^1, \dots, X^k) \subset \mathbb{R}^{n \times (d_1 + \cdots + d_k)}$ varies within each class. More explicitly, we use the metric:

$$\text{Error} = \frac{1}{n} \sum_{\text{classes } \mathcal{C}} \sum_{x \in \mathcal{C}} \|x - \text{mean}\,(y \in \mathcal{C})\|. \quad (10)$$

Where the norm $\|\cdot\|$ is the same as defined in III-C. The results for each of our sample methods are given in table I. Most of the experimental methods are pictured above, with the exception of our multimodal method with weights drawn from the standard 2-norm. This is a slight variation on the method explained in III, where instead of the norm used in III-C, we define

$$\|x\| = \sqrt{\frac{\|x^1\|^2}{\lambda_1} + \cdots + \frac{\|x^k\|^2}{\lambda_k}}, \quad (11)$$



(a) DFC optical data

(b) DFC lidar data

(c) Optical-only segmentation

(d) Lidar-only segmentation

Fig. 2: Single-modality classifications.

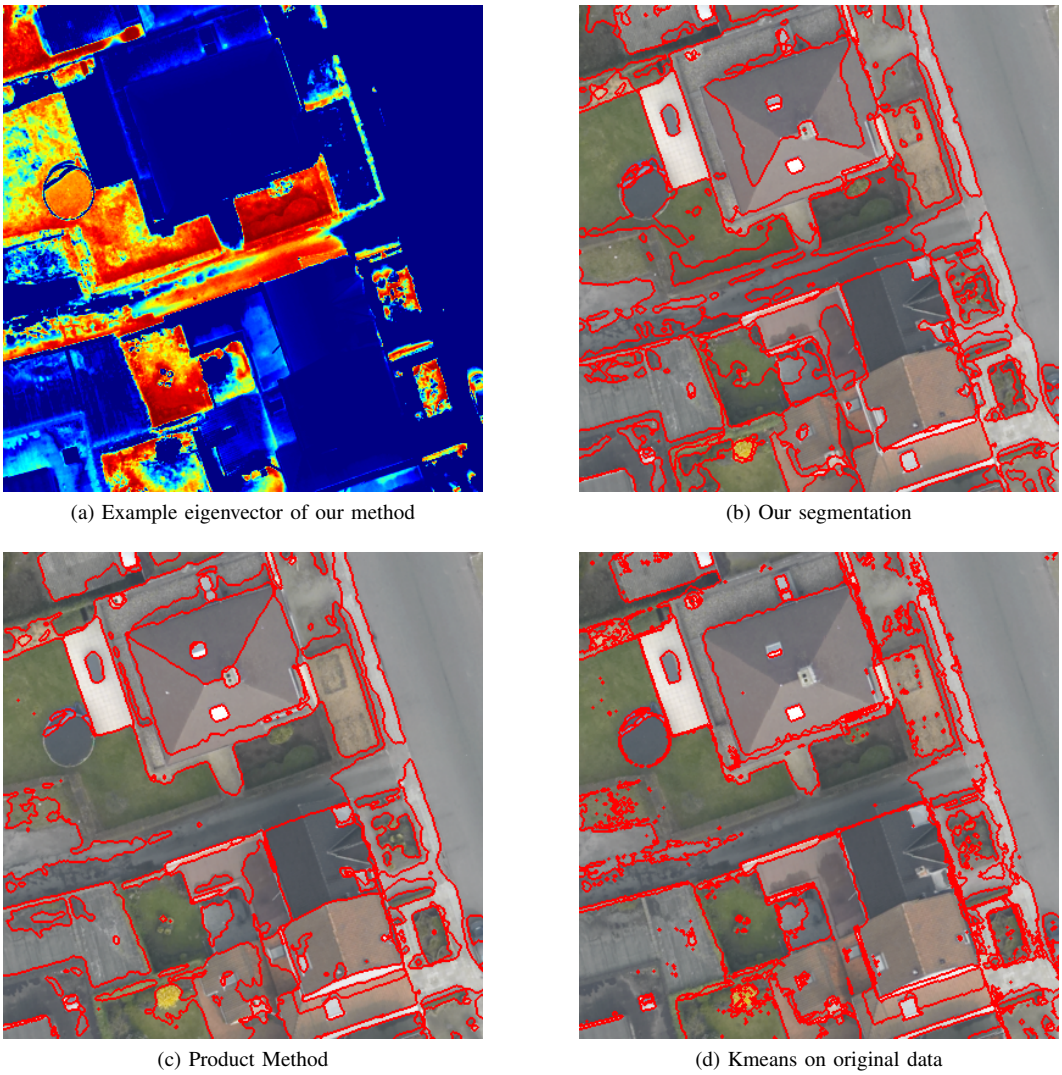
and create the weight matrix using this metric. Interestingly, the change in norm causes very little change in the final segmentation. The runtimes here are taken from a personal laptop, and should only be used for relative comparison within this paper.

Method	**Error**	**Runtime**
Our method 3b	0.40	TODO:
Our method with 2-norm (eqn 11)	0.41	need
Product method 3c	0.43	to
Graph cut on optical only 2c	0.83	update
Graph cut on lidar only 2d	0.75	this
K-means on original data 3d	0.75	table

TABLE I: DFC quantitative comparisons

B. Umbrella Data

In fig 4 we show the results of our method applied to another optical/lidar set (found in [29]), 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 classification 4d can be understood by looking at the individual feature vectors. In 4c we show the vector responsible for separating the umbrellas from the background wall (using the lidar data), as well as from the black umbrella stands (using the RGB data).



(a) Example eigenvector of our method

(b) Our segmentation

(c) Product Method

(d) Kmeans on original data

Fig. 3: Various segmentations

Once again, we present comparisons with other methods both visually (figs 4e, 4f) and quantitatively (table II).

Method	**Error**	**Runtime**
Our method	0.31	9.0s
Our method with 2-norm	0.33	8.4s
Product method	0.51	16.6s
Graph cut on optical only	0.86	8.1s
Graph cut on lidar only	0.50	6.7s
K-means on original data	0.52	3.1s

TABLE II: Umbrella data quantitative comparisons

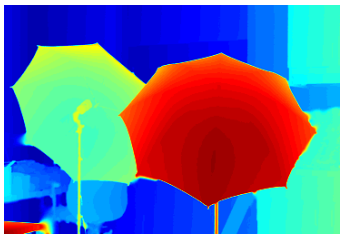
C. Jade Plant Data

Found in the same paper as the umbrella data [29], we test our method against another optical/lidar scene of a jade plant. In figure 5 we once again display the results of several different classification methods. This time we also show additional eigenvectors extracted from the graph Laplacian, to give a better idea of how the final segmentation is obtained. Here 5c is responsible for separating the black background from the rest of the image. 5d differentiates the foreground plant and blue square from the boxes piled behind. And 5e adds some finer details to the segmentation by pulling out the blue square in the foreground, the ceramic pot, and showing a difference between the plant leaves and the stems.

V. Conclusions

In conclusion, graph-based methods provide a straightforward and flexible method of combining information from multiple datasets. By defining a weight map $\mathbb{R}^{n\times(d_1+\cdots+d_k)} \to \mathbb{R}_{\geq 0}$ with some reasonable norm-like

(a) Umbrella optical data

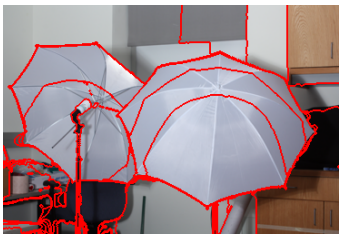


(b) Umbrella lidar data

(c) Example eigenvector

(d) Our segmentation

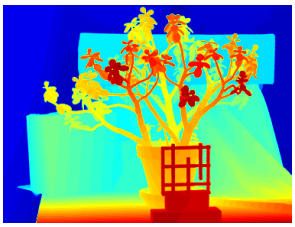
(e) Product Method



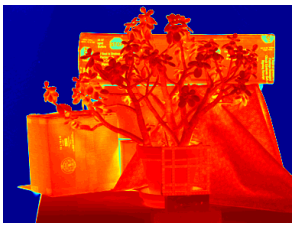
(f) Kmeans on original data

Fig. 4: Umbrella data results

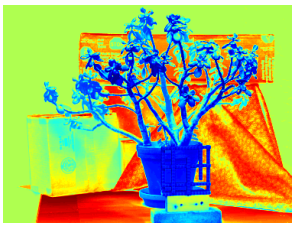
(a) Jade plant optical data



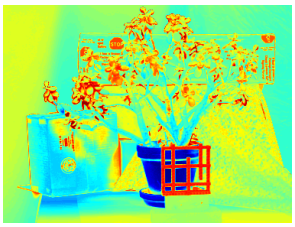
(b) Jade plant lidar data



(c) Example eigenvector 1



(d) Example eigenvector 2



(e) Example eigenvector 3

(f) Our segmentation

(g) Product Method

(h) Kmeans on original data

Fig. 5: Jade plant data results

properties, we can 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. However this portion of our method could easily be replaced with a more in-depth approach, such as a Mumford-Shah model [20], or even a semi-supervised method such as [22]. Our next area of interest is the removal of the co-registration assumption. In section IV our two images are of the same underlying scene, where pixels correspond exactly between images. We could not, for example, process two images taken from different angles. Our goal for the future is to remove this restriction and develop an algorithm that can be applied more varied datasets.

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)

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. (document), 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] C. Pohl and J. L. Van Genderen. Review article multisensor image fusion in remote sensing: Concepts, methods and applications. *International Journal of Remote Sensing*, 19(5):823–854, 1998. 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, III-A2, V

[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-B

[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-A2, III-B, V

[23] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007. II, III-A, III-A2

[24] 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-A1

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

[26] 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-B

[27] 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-B

[28] 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. IV-A

[29] 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