Multimodal Data Comparison via Graph Matching

Geoffrey Iyer, Jocelyn Chanussot, Andrea L. Bertozzi

1 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 [1, 2]. Similarly, in medicine, the authors of [3] and [4] fuse the results of two different types of brain imaging to create a final image with better resolution than either of the originals. However, correctly processing a multimodal dataset is not a simple task [5]. The main difficulty in multimodality lies in finding a way to coordinate information that is represented in different formats. In the current state-of-the-art, most multimodal algorithms apply only to one specific problem. Therefore it is desirable to produce an algorithm that is robust to many different types of input data.

One way of addressing multimodality in a general sense is to view each set of data as a manifold in some $\mathbb{R}$-space. We then look to form comparisons between the different manifolds by using topological/geometric qualities of the sets. Since each manifold is derived from the same source, albeit through different sensors, we expect that there is enough commonality between the sets to obtain a reasonable correspondence this way. Some examples of this approach can be found in [6], where the authors use Canonical Correlation Analysis to compare the sets, and in [7, 8], both of which use graph-based methods to perform feature extraction, then make comparisons on the feature level.

In this project, we aim to create an algorithm to compare manifolds using graph matching techniques. Each dataset will be represented as a weighted graph (similar to [7, 8]), which we will then use for the remainder of the algorithm. The advantage of the graph representation is its robustness to deformation. The weights are based only on the similarity between graph nodes, so they are invariant under conformal mappings, and relatively resilient towards continuous change. We will then solve a registration problem, matching nodes between the two graphs in a manner that preserves the graph geometry. This match can then be lifted to the original datasets, giving us a tool with which compare the different modalities. In the current state of the project, we have created a preliminary implementation of a graph matching algorithm based on the work in [9, 10] and have completed a few experiments to test (a) the quality of the match, and (b) our ability to use this match to solve interesting problems involving more than one dataset.

The rest of this paper is organized as follows. In section 2 we present the theory behind our algorithm. Section 2.1 defines the weighted graph matching problem (WGMP) in detail. We then reprise the results of [9, 10] in 2.2, followed by our own improvements in 2.3. In section 3 we show the efficacy of our algorithm on an example dataset, followed by an application of our graph matching towards a change detection problem. Finally, in 4 we expound upon the direction of this project, explaining our main goal, as well as several additional contributions that we aim to make.

2 The Method

Given two datasets X, Y, we will define the corresponding undirected, weighted graphs $G = (X, W_X)$, $H = (Y, W_Y)$. Here the nodes of G (resp. H) correspond to the points in X (resp. Y). The weight matrix W_X has size $|X| \times |X|$, and the value $W_X(i, j)$ represents the similarity between nodes $X(i), X(j)$ (and similar for W_Y). There are many possible definitions of W_X in the literature, but the most common involve a RBF kernel, as we use in our experiments (section 3, equation 10).

2.1 The Weighted Graph Matching Problem (WGMP)

Let $G = (X, W_X)$, $H = (Y, W_Y)$ be undirected, weighted graphs. Here X, Y represent the nodes of G, H (respectively), and W_X, W_Y are the corresponding matrix of edge weights. For convenience of notation in the statement of the problem, we will assume $|X| = |Y| = N$. The extension to the general case is quite straightforward, and will be explained in section 2.2. The goal of the Weighted Graph Matching Problem is to find a bijection $\rho : X \to Y$ that minimizes the squared difference of edge weights:

$$\text{argmin}_\rho \sum_{i=1}^{N} \sum_{j=1}^{N} \left(W_X(i,j) - W_Y(\rho(i), \rho(j))\right)^2 . \tag{1}$$

It is often easier to view the map ρ as a permutation of the indices $\{1, 2, \ldots, N\}$. Let P be the corresponding $N \times N$ permutation matrix. Then we look to minimize

$$\text{argmin}_P \left\| P W_X P^T - W_Y \right\|_F^2 . \tag{2}$$

Finding an exact solution to this problem is NP-Hard [16]. Instead, we look for an approximate solution via the methods presented in [9, 10]. In particular, we create a feature space in which our two graphs can be more directly compared, and perform the matching there.

2.2 Solving the relaxed WGMP via the graph Laplacian

As the WGMP is too difficult to solve exactly, we instead introduce a relaxed version of the problem which is much more reasonable. Specifically, instead of choosing P a permutation matrix as in 2, we look for an orthogonal matrix

$$Q^* = \text{argmin}_{QQ^T = I} \left\| Q W_X Q^T - W_Y \right\|_F^2 . \tag{3}$$

This problem was solved theoretically in [9] using eigenvectors of the graph Laplacian. Specifically, for each graph G_k ($j \in \{X, Y\}$) we define the graph Laplacian

$$L_k = D_k - W_k, \tag{4}$$

where D_k is the diagonal matrix of degrees,

$$D_k(i,i) = \sum_{j=1}^{n} W_k(i,j). \tag{5}$$

Note that L_k is symmetric, and is in fact positive semidefinite [17]. Let $L_k = U_k \Lambda_k U_k^t$, be the eigendecomposition of the Laplacian. Then the spectral graph matching theorem from [9] states that if each L_x, L_y has distinct eigenvalues, the optimal Q from 3 is given by

$$Q^* = U_Y^T S U_X, \tag{6}$$

where S is a diagonal matrix with values ± 1 to account for the sign ambiguity in eigenvectors. The calculation of S can be rather frustrating, as there are 2^N possibilities for this matrix. In [9] the authors omit S entirely by replacing U_k with $|U_k|$ in equation 6. In [10] the authors determine S by looking at the histograms of the columns of U_k (which are invariant under permutation) and adding ± according to the best fit. In the current state of our project, we are using a semi-supervised method to determine S. If we know of even one preexisting match between the nodes in X, Y, we can use this to compare eigenvectors U_X, U_Y and add the appropriate signs.

To complete our approximate solution to the WGMP, we use the matrix Q^* to find a matching

$$\rho : \{1, 2, \ldots, N\} \to \{1, 2, \ldots, \}. \tag{7}$$

In the original formulation of the WGMP 2, an entry of 1 at position (i, j) in the permutation matrix P represents a match between node i of Y with node j of X. In the relaxed version of the problem, $Q^*(i, j)$ gives the similarity between node i of X and node j of Y rather than an exact matching. But the task of converting Q^* to a permutation ρ is easily solved using the Hungarian algorithm, which in $O(N^3)$ time finds the matching ρ that maximizes

$$\sum_{i=1}^{N} Q^*(i, \rho(i)). \tag{8}$$

In practice, one often does not use every eigenvector of L_X, L_Y in the calculation of Q. Rather, we choose a number $K \ll N$, and let U_k be the $N \times K$ matrix where the columns are the eigenvectors corresponding to the K smallest eigenvalues. Heuristically, the columns of U_k represent features extracted from the original graph, and the size of the corresponding eigenvalues represents the strength of each particular feature. If we then assume that the sign ambiguity has been dealt with (so that $S = I$), then $Q^* = U_Y U_X^t$. That is, $Q^*(i, j)$ is the dot product of row i of U_Y with row j of U_X. So we are determining match strength by comparing the images of our data in this new feature space. In doing this, we can also handle the case where $|X| \neq |Y|$. If we chose $K \leq \min(|X|, |Y|)$, then formula 6 will still hold, and Q^* will be an $|Y| \times |X|$ matrix representing the node similarities.

2.3 Hierarchical graph matching

The major benefit of using the Hungarian algorithm in 2.2 is that it results in the optimal one-to-one matching based on the input data, but the $O(N^3)$ runtime presents a major problem when dealing with larger datasets. To address this, we introduce a hierarchical algorithm for graph matching that solves the problem by making many smaller matches, thereby circumventing the N^3 issue. The key step in the hierarchical matching algorithm is the creation of smaller "coarsified" graphs $\tilde{G}, \tilde{H}$ of size $M \ll N$. The goal is for $\tilde{G}, \tilde{H}$ to represent the same geometrical structure as G, H, with significantly reduced size. In the current state of the project we are still investigating different methods for creating $\tilde{G}, \tilde{H}$, but the overarching idea is to choose a few nodes of high degree in the original graphs G, H to serve as representatives for larger clusters. These representatives will then form the $\tilde{G}, \tilde{H}$ that we desire.

Once we have created the coarsified graphs $\tilde{G}, \tilde{H}$, the hierarchical matching algorithm is as follows. We first run the standard graph matching algorithm on $\tilde{G}, \tilde{H}$. Then, for each match $i \to j$ in the coarse graphs, we run our graph matching algorithm between the corresponding clusters in the original graph. So in total, we create 1 match of size M, and M matches of size $\frac{N}{M}$, which gives a new runtime of $O\left(M^3 + \frac{N^3}{M^2}\right)$. This is significantly faster than the original $O(N^3)$ algorithm, and allows us to work with much larger datasets. In practice, the code takes approximately 15 seconds to run when $N = 10,000$ and $M = 50$. In theory, we could achieve an even better runtime for the matching process by performing multiple coarsification steps, and iteratively matching the graphs on increasingly fine levels. However, this would not significantly affect the runtime of the full algorithm, as finding the eigenvectors of L_X, L_Y would then become the bottleneck for the problem.

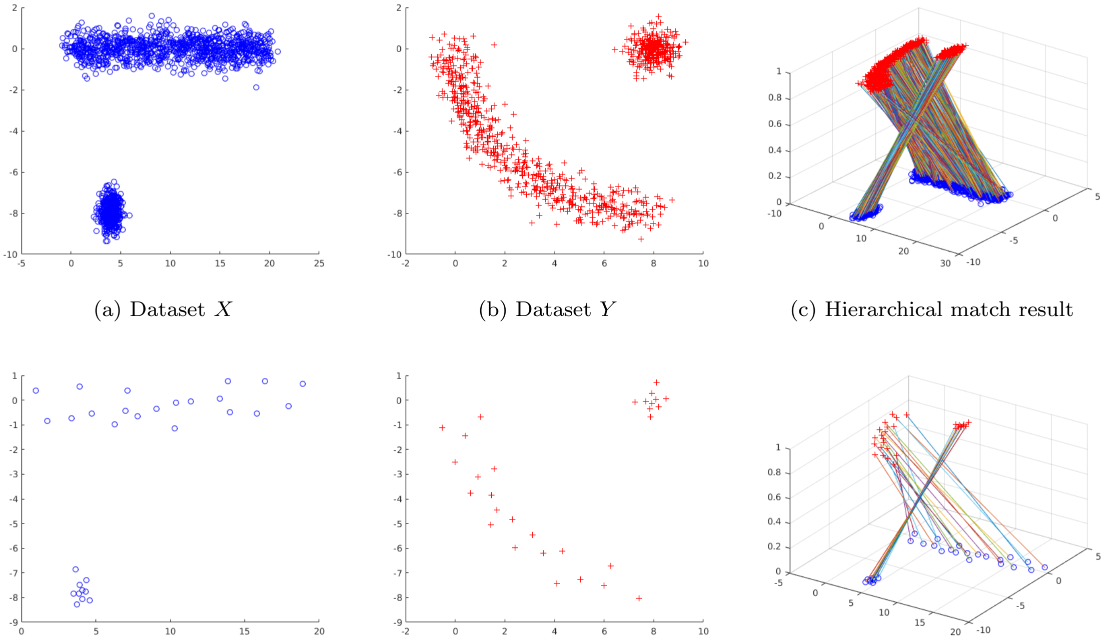
3 Experiment

3.1 Graph matching

Here we show the results of our graph matching algorithm on a synthetic dataset, which is pictured in figure 1. Here the nodes of the graphs G, H are represented by 2-dimensional datasets X, Y, and the weight matrices W_X, W_Y are determined via an RBF kernel applied to the 2-norm.

$$E_X(i, j) = \|X(i) - X(j)\| \tag{9}$$

$$W_X(i, j) = -\exp\left(\frac{E_X(i, j)}{std(E_X)}\right), \tag{10}$$



(a) Dataset X

(b) Dataset Y

(c) Hierarchical match result

(d) Coarsified data $\tilde{X}$

(e) Coarsified data $\tilde{Y}$

(f) Graph match on coarse data

Figure 1: Example hierarchical matching on synthetic data

and similar for W_Y. The matching is then calculated using the hierarchical algorithm in 2.3, with the match on the coarse level begin shown in figures 1d, 1e, 1f. This example has datasets of size $N = 1500$, with $M = 50$, so that the coarsified data has size $\left|\tilde{X}\right| = \left|\tilde{Y}\right| = 30$.

The purpose of this example is to show that the matching algorithm can recognize similar shapes in data that has been altered in a smooth-enough manner. Both sets X, Y contain a tight cluster of points, as well as longer line segment. In figures 1c, 1f we see the final result of the algorithm, where each match is represented by a line connecting the two points in question. As we can see in the figure, the algorithm successfully matches the objects based on their shape, as we desired.

3.2 Change detection using graph matching

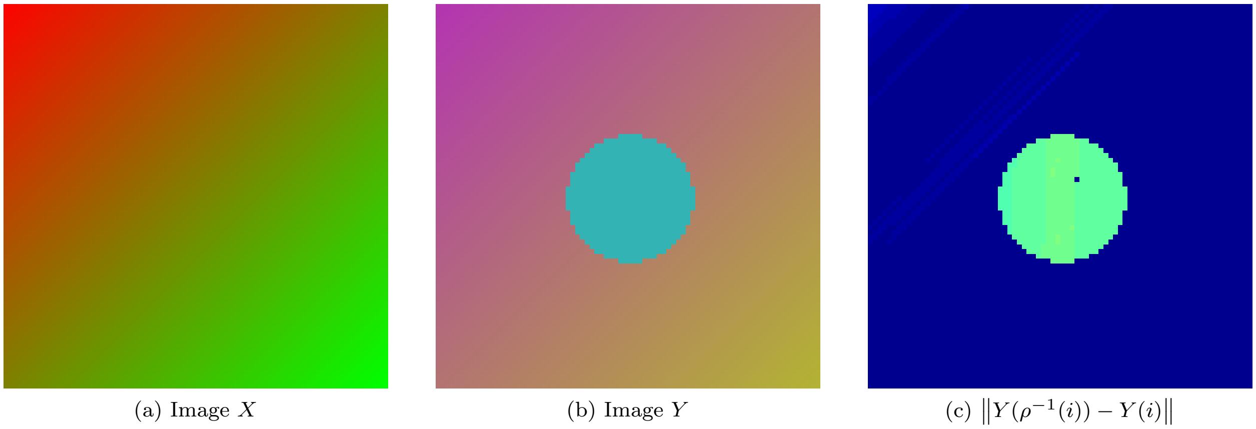
One possible application of graph matching is in change detection. Suppose that the sets X, Y represent two images of the same scene, taken at different times. A direct comparison of X to Y is often not useful, as it is possible for individual pixels to change drastically while keeping the overall structure of the image the same. For example, if the images X, Y were captured in different lighting, then the set $X - Y$ would show many "false" changes. Instead, we turn to graph matching as a way to compare pixels between X, Y in the context of the larger image.

We will track the changes as follows: apply the matching algorithm to X, Y, and let $\rho : \{1, 2, \ldots, N\} \to \{1, 2, \ldots, N\}$ be the permutation of the indices corresponding to the match $X \to Y$. Then we measure the degree of change in the scene at pixel i via the quantities

$$\text{Change in } X \text{ at pixel } i = \|X(i) - X(\rho(i))\| \tag{11}$$

$$\text{Change in } Y \text{ at pixel } i = \left\|Y(\rho^{-1}(i)) - Y(i)\right\| \tag{12}$$

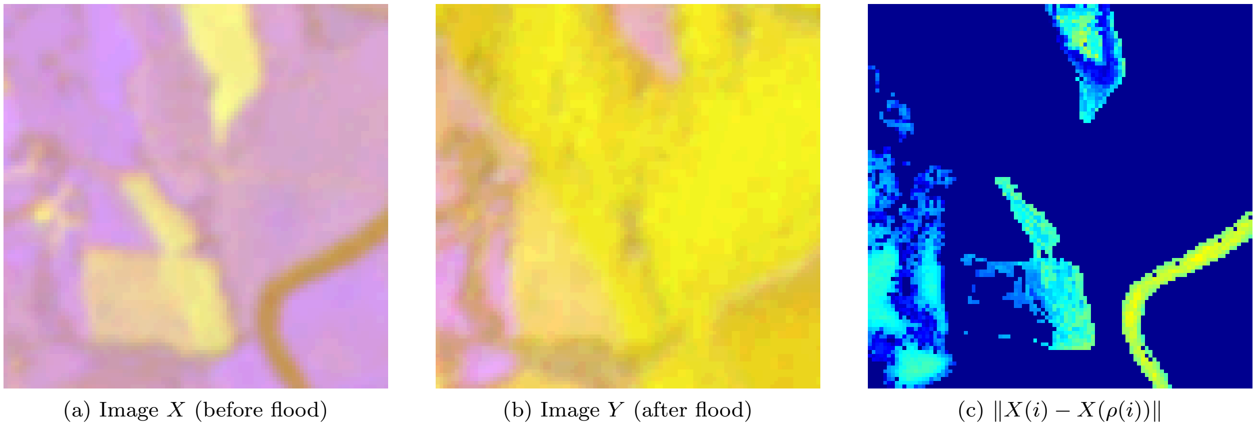
As a first example, we apply the algorithm to a synthetic dataset, with the results shown in figure 2. This data was constructed to mimic the above discussion, showing both a change in overall hue between images X, Y, as well as an actual structural change with the addition of a blue dot in Y. In this example, we


(a) Image X

(b) Image Y

(c) $\|Y(\rho^{-1}(i)) - Y(i)\|$

Figure 2: Example change detection on synthetic data


(a) Image X (before flood)

(b) Image Y (after flood)

(c) $\|X(i) - X(\rho(i))\|$

Figure 3: Example change detection on DFC 2010 data

consider the dot in Y to be the only true "change" between the images. The change in hue represents some incidental affect in the data capture process, and should be ignored by the algorithm. Here, both images are size 80×80, for a total of $N = 6400$ pixels. The algorithm runs in roughly 10 seconds, and successfully highlights the dot in Y, as desired.

At this point in the project, we began working with real-world datasets. In figure 3 we show a preliminary result of our algorithm on data from the 2010 Data Fusion Contest [18]. The dataset consists of remote sensing image of Gloucester, UK, taken both before (3a) and after (3b) a flood in the year 2000. The final result in figure 3c is difficult to fully interpret, but it is encouraging to see the algorithm highlighting noteworthy shapes from the individual images. As we continue to improve our methods, we expect to see even more obvious results.

4 Future Work

Our main goal for this project is to improve and expand on the robustness of our algorithm. As stated in the introduction, a flexible method for comparing data from different modalities could be useful in a large variety of situations. Furthermore, we will be working towards a more complete implementation of change detection using graph matching. When working with multimodal data, we expect significant redundant information between the different sources. However, we do not expect the modalities to be entirely redundant. Each source should contain some unique information that can contribute to our understanding of the source object.

We wish to use our change detection algorithm (or some other method like it) to highlight this information, allowing users to quickly sort out the key information that each modality brings.

There are also many possible applications of this work that we have not yet explored. Having a matrix Q^* (defined in 2.2) with precise numbers representing the similarity between graph nodes gives us many options in dealing with our data. For example, we could fuse two datasets by identifying matches with a high similarity score (and leaving matches with low similarity as separate).

Lastly, we may be able to optimize our hierarchical matching algorithm to get an improvement on the state-of-the-art for graph matching (an not the applications thereof). In comparing our work to some recent methods [10, 11, 12, 13, 14] it appears that our hierarchical algorithm results in a respectable runtime. With a more careful treatment of the eigenvectors of the graph Laplacian, as in [19, 20, 21], we may be able to create a graph matching algorithm that processes data significantly faster, while maintaining roughly the same degree of accuracy.

References

[1] 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. 1

[2] 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. 1

[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. Integr. Neurosci.*, 11(3):313–337, Sep 2012. 1

[4] 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. 1

[5] 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. 1

[6] Yi-Ren Yes, Chun-Hao Huang, and Yu-Chiang Frank Wang. Heterogeneous domain adaptation and classification by exploiting the correlation subspace. *IEEE Transactions on Image Processing*, 23(5):2009–2018, May 2014. 1

[7] Chang Wang and Sridhar Mahadevan. Manifold alignment preserving global geometry. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, pages 1743–1749. AAAI Press, 2013. 1

[8] Devis Tuia and Gustau Camps-Valls. Kernel manifold alignment. Mar 2015. 1

[9] S. Umeyama. An eigendecomposition approach to weighted graph matching problems. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(5):695–703, September 1988. 1, 2.1, 2.2, 2.2, 2.2

[10] David Knossow, Avinash Sharma, Diana Mateus, and Radu Horaud. *Inexact Matching of Large and Sparse Graphs Using Laplacian Eigenvectors*, pages 144–153. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. 1, 2.1, 2.2, 4

[11] H. A. Almohamad and S. O. Duffuaa. A linear programming approach for the weighted graph matching problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(5):522–525, May 1993. 4

[12] JT Vogelstein, JM Conroy, V Lyzinski, LJ Podrazik, SG Kratzer, and ET Harley et al. Fast approximate quadratic programming for graph matching. *PLoS ONE*, 10(4), 2015. 4

[13] Marcelo Fiori, Pablo Sprechmann, Joshua T. Vogelstein, Pablo Muse, and Guillermo Sapiro. Robust multimodal graph matching: Sparse coding meets graph matching. In C.j.c. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 127–135. 2013. 4

[14] M. Zaslavskiy, F. Bach, and J. P. Vert. A path following algorithm for the graph matching problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2227–2242, Dec 2009. 4

[15] D. CONTE, P. FOGGIA, C. SANSONE, and M. VENTO. Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(03):265–298, 2004.

[16] Vikraman Arvind, Johannes Köbler, Sebastian Kuhnert, and Yadu Vasudev. *Approximate Graph Isomorphism*, pages 100–111. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. 2.1

[17] Bojan Mohar. The laplacian spectrum of graphs. *Graph Theory, Combinatorics, and Applications*, 2:871–898, 1991. 2.2

[18] Nathan Longbotham, Fabio Pacifici, Taylor Glenn, Alina Zare, Michele Volpi, Devis Tuia, Emmanuel Christophe, Julien Michel, Jordi Inglada, Jocelyn Chanussot, and Qian Du. Multi-modal change detection, application to the detection of flooded areas: outcome of the 2009-2010 data fusion contest. *IEEE J. Sel. Topics Appl. Earth Observ.*, 5(1):331–342, Feb. 2012. 3.2

[19] 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. 4

[20] 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. 4

[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. 4