A Study of Graph Cuts Theoretic Concepts with Digital Signal Processing with Mathematical Concepts

Dayanand and Arihant Jain

Department of Mathematics, Dr. A. P. J. Abdul Kalam University, Indore (M.P.), India-452016

Article Info
Page Number: 1359-1369
Publication Issue:
Vol. 71 No. 4 (2022)

Article History
Article Received: 25 March 2022
Revised: 30 April 2022
Accepted: 15 June 2022
Publication: 19 August 2022

Abstract

Optimization of variety of objective functions using graph cuts in different move spaces is studied in detail in the paper. The study concludes that, an objective function can be minimized using graph cuts provided it is FNO-optimizable. Characterizations of two classes of FNO- optimizable functions (O^2 and O^3) are given and many mathematical results in this regard are proved in the paper. These characterizations contribute in easily identifying the image processing problems which can be addressed through graph cuts notion. However, further exploration of the concepts studied/ defined in the paper is required to strengthen the designed mathematical framework. Due to programming limitation on our part, we could not construct a computationally time-effective computer program encoding the graph-cuts model. There is a good scope of improvement in the implementation of graph theoretic models designed in the present work on appropriate programming platform.

Keywords: Theoretical DSP, graph Cut.

1 PRELIMINARIES

1.1 Graph Cuts for Binary Optimization

As a binary optimization technique, graph cuts will be discussed in this section. In reality, minimum -cut/maximum-flow algorithms are essentially binary techniques, as they are based on the concept of minimization. Binary issues, then, are the most fundamental example of graph cutting.

1.2 An Introduction to Mathematical Image Processing

In addition to medical imaging, astronomy, astrophysics and surveillance, image processing is also used in video compression and transmission. Signals are images in one dimension, whereas images in another dimension are termed images. We deal with planar images in two dimensions, and volumetric images in three dimensions (such as MR images). Alternatively, they can be colors images or grayscale images (single-valued functions) (vector-valued functions). As a result of noise, blur, and other flaws, captured images are typically deteriorated. Preprocessing is thus required before any further analysis and feature extraction can be conducted on these pictures.

While taking this course students learn how to mathematically express the following processes: denoising and deblurring of images; augmentation of images; and edge detection.

As far as spatial filtering is concerned there will be incomplete imitative, gradients, the Laplacian or their separate estimates by limited dissimilarities, averaging filters and order figures filters as well as difficulty. As far as the incidence domain is concerned, there will be Fourier transforms, low-pass and high-pass filters, as well as zero-crossings of the Laplacian, among others. A weighted Laplace equation for image restoration, or the use of snakes for image segmentation, can be studied if you have time.

2 STATISTICAL JUSTIFICATION OF THE APPROACH

In the study, we attempt to solve the problem of pixel labeling by objective function minimization approach. At the first glance, the approach seems to be a deviation from the main goal but, it can be defended by Baysian statistics. In this section, prerequisites of Baysian statistics have been briefly discussed.

3 GRAPH CUT MODEL FOR UNIFORMLY SMOOTH STRUCTURE

In this model, the structure constraint is encoded by the sub - function $\Psi_{v,w}(x_v, x_w)$ as linear relationship of neighbouring pixels with corresponding weights. It is defined by $\Psi_{v,w}(x_v, x_w) = c(v,w)\left|x_v - x_w\right|$. Where $c(v,w)$ is the constant corresponding to pair of pixels v and w. With the expression of structural constraint defined as above, the objective function to be minimized takes the form,

$$O(X) = \sum_{v} \varphi_v(x_v) + \sum_{\{v,w\}\in N} c(v,w)|x_v - x_w| \qquad (3.1)$$

The graph cuts can efficiently minimize the objective function (3.1) globally.

3.1 Construction of Network

The graph cuts technique separates a collection of pixels into two subgroups and hence determines one of the most cost efficient binarization on the given group of pixels in the light of objective function. Let $\{v_1, v_2, \ldots\ldots, v_n\}$be set of pixels and $\{\sigma_1, \sigma_2, \ldots, \sigma_p\}$ be set of all possible labels. For the problem under consideration, we construct a graph G with vertex set V containing (n_p - n +2) vertices and edge set E. To be more specific, $V = \{v_{11}, v_{12}, \ldots, v_{1(p-1)}, v_{21}, v_{22}, \ldots, v_{2(p-1)}, \ldots, v_{n1}, v_{n2}, \ldots, v_{n(p-l)}, s, t$with two terminal vertices s and t called source and sink. Corresponding to each pixel v_i $(1 \le i \le n)$, there are (p - 1) vertices $v_{i1}, v_{i2}, \ldots\ldots\ldots, v_{i(p-l)}$ in the graph. The edge set $E = \left\{e_{ij}^t (1 \le i \le n, 1 \le j \le p), e_{ijk}^n (1 \le i \le n, 1 \le j \le n, 1 \le k \le p-1)\right\}$ consists of two types of edges: e_{ij}^t called terminal edges and e_{ijk}^n called non-terminal edges. $e_{ij}^t (2 \le j \le p-1)$ is an edge connecting $v_{i(j-1)}$ and v_{ij} whereas e_{ijk}^n is an edge connecting v_{ik} and v_{jk} . e_{i1}^t is an edge connecting s and v_{i1} whereas e_{ip}^t is an edge connecting $v_{i(p-l)}$ and t. Every non-terminal edge e_{ijk}^n has a weight $c(v_i, v_j)$ and terminal edge e_{ij}^t carries a weight $A_i + \varphi_i(\sigma_j)$, where A_i is a constant with $A_i > (p-1)\left(\sum_{v_j \in N_{v_i}} c(v_i, v_j)\right)$. Figure 3.1 shows a sub-network of G representing the structure of the network for two neighbouring pixels v_i and v_j.

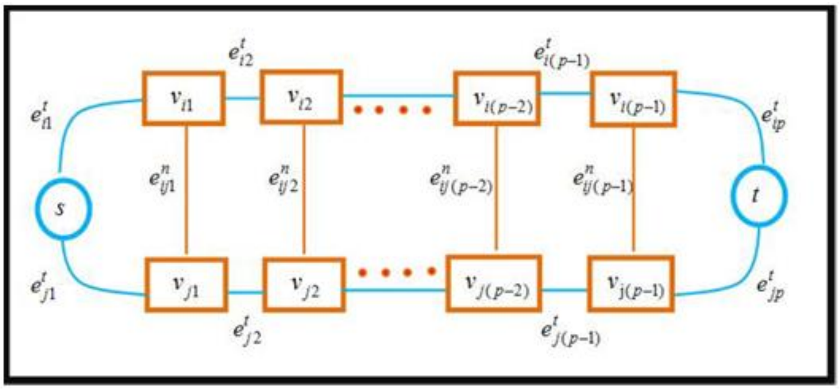

Figure 3.1 Sub-network of G for neighboring pixels v_i and v_j of pixel the pixel set of image

4. DEFINITION OF THE PROBLEM TO BE ADDRESSED

We transform the problem of image binarization into an equivalent optimization problem and attempt to solve it through network flow terminology (i.e. Graph cuts). Max flow min- cut theorem given by Ford and Fulkerson plays a very crucial role in the model as in all graph cuts models.

A digital image of text document (i.e. scanned text) is composed of rectangular arrangement of pixels, each of which represents intensity. Binarization is a process of defining a function $X: V \to \{t', b\}$ (where V is the set of all pixels of the textual image) which reassigns a binary value t' (Text) or b (Background) to every pixel v of the image, subject to the given data. The choice of the function depends on two constraints. 1) The neighboring pixels should be assigned similar value by the function almost everywhere with the exception of the pixels representing boundary of the text. 2) Theassignment should be made in light of the data given by scanned image. We have employed these constraints to construct the function. The first step includes construction of an objective function considering the constraint of the problem. Let V be a set of all pixels of the image and $\{g_v: v \in V\}$ be the set of grey values of pixels. The objective function $O: \Omega \to \mathbb{R}$ provides the measure of inappropriateness for every possible binarization function X from Ω and plays a crucial role in the selection of most suitable function

$$i.e.\, O(X) = \sum_{\substack{(u,v)\in N \\ u,v\in V}} kp_{uv} + \sum_{v\in V} |X_v - g_v| \qquad (4.1)$$

4.1 Theoretical Justiftcatton of the Model

In this section, we will prove that, the model efficiently minimizes the objective function and produces the binarization of the image which has minimum value under objective function (4.1). This also proves the superiority of the model among prevailing binarization techniques in light of the objective function.

Theorem 4.1.1

For any scanned textual image, the set Ω of all corresponding binarization functions and the set C of all cuts on the network flow corresponding to the image are in one to one correspondence.

Proof: Let $X: V \to \{t', b\}$ be a binarization function corresponding to the given scanned image. Then, there exists a cut $C = \{S, T\}$ on the network flow corresponding to the image defined as follows:-

$$S = \{v | X_v = 0\} \; and \; T = \{v | X_v = 255\}$$

Note that, $X_v = 0$, when $X(v) = t'$ and $X_v = 255$ when $X(v) = b$.

This proves that, every binarization function gives rise to a cut on the network flow constructed forthe textual image under consideration.

Conversely, let $C = \{S, T\}$ on the network flow for the given textual image. Then, we can define X as follows:

$$X(v) = t', if \; v \in S$$

$$X(v) = b, if \; v \in T$$

This proves the theorem.

THEOREM 4.1.2

The minimum cut on the network flow constructed for the textual image gives binarization which minimizes the objective function.

Proof: let X be the binarization corresponding to any cut C of the network flow. First, we show that, cost of the cut C is O(X).

Note that, cost of any cut is sum of weights of the edges which are member of the cut set. In case of our network flow, there are two types of edges: (i) non-terminal edges e_{uv}^n joining neighboring vertices u and v of the network flow (ii) Terminal edges e_v^s and e_v^t connecting vertex v with the terminal vertices s and t respectively.

Thus, the cost of C is given by,

$$|C| = \sum_{\substack{(u,v)\in N \\ e_{uv}^n \in C}} |e_{uv}^n| + \sum_{e_u^s \in C} |e_u^s| + \sum_{e_u^t \in C} |e_u^t| \qquad (4.2)$$

4.2 Implementation and Results

The model is implemented using Java programming language. The input image is first converted into grayscale in order to facilitate the computation. The initial binary labeling is obtained through thresholding. Then after, the network flow is constructed, where weights of the edges are decided based on the binary value assigned to a particular pixel by the initial labeling. The code finds the most cost effective (with reference to cost/value/penalty assigned by the objective function) possible binary labeling in the move space. Following is the code for implementation of the model in Java programming language:-

```
/******************************************************************************
* Compilation: javac FordFulkerson3.java
* Execution: java FordFulkerson3 V E
* Dependencies: FlowNetwork.java FlowEdge.java Queue.java
*
* Ford-Fulkerson algorithm for computing a max flow and
* a min cut using shortest augmenthing path rule.
*
******************************************************************************/
import java.awt.image.BufferedImage;
import java.awt.image.Raster;
import java.awt.image.ColorModel;
import java.io.IOException;
import java.io.File;
import java.io.PrintWriter;
import javax.imageio.ImageIO;
public class FordFulkerson3 {
	private boolean[] marked; // marked[v] = true iff s->v path in residual graph
	private FlowEdge[] edgeTo; // edgeTo[v] = last edge on shortest residual s->v path
	private double value; // current value of max flow
```

The results obtained through the code are given below:-

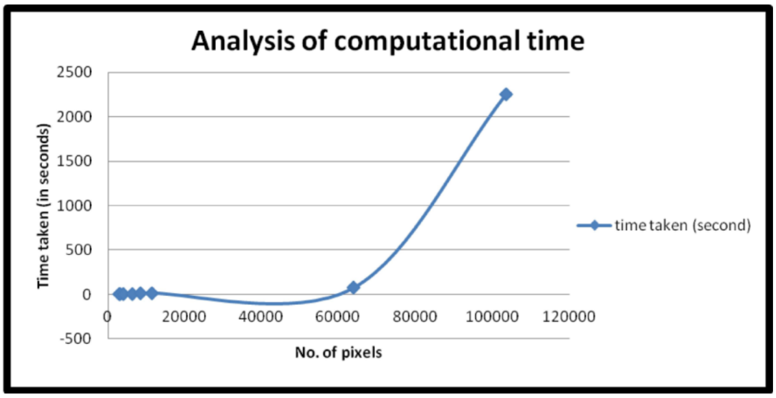

Figure 4.1 Various image sizes versus Time required for binarization

As discussed earlier, the code turns out be quite expensive in terms of time it takes to handle the image for binarization, especially for larger image segments. Table 4.1 presents the time taken by the code for the image segments of various dimensions. For images of more than one tenth of a million pixels, the code takes about 38 minutes, which is very high time when compared to other popular binarization methods. Methods based on global threasholding take comparatively very small time for images of same size. Figure 4.1 shows that the time required for binarization of the image using our code grows exponentially with image size after a fixed stage. This is very serious drawback of our model. However, this deficiency could be easily overcome by better coding. Due to our limited programming skills, the code we wrote has vast scope for improvement.

Table 4.1 Time required for binarization for various image sizes

Dimension of the scanned image	No. of pixels in the image	Time taken (in Seconds)
91×33	3003	2
91×35	3185	2

89×45	4005	3
100×63	6300	5
92×92	8464	10
120×95	11400	15
320×200	64000	75
480×216	103680	2258

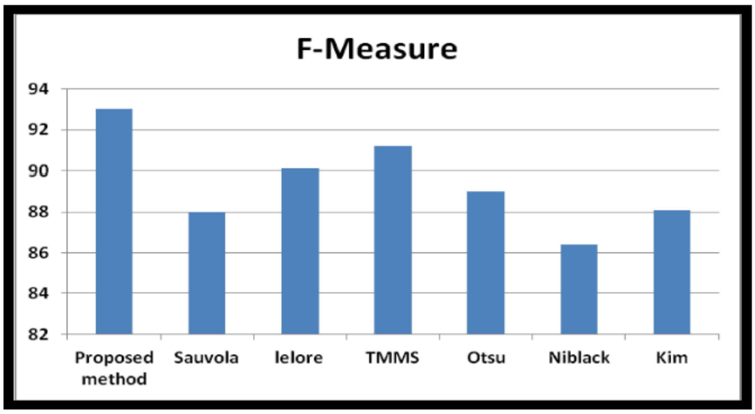

Figure 4.2 Bar graph showing various binarizatin methods and corresponding F-measure

There are many evaluation techniques to measure the quality of the binarization based on two main approaches: (1) pixel based accuracy evaluation and (2) OCR based evaluation. We have used some of the most prevalent evaluation techniques to measure the quality of our binarization results. The results are presented in the table 4.2.

We have used PERR, MSE and F-measure as evaluation techniques to measure the quality of binarization.

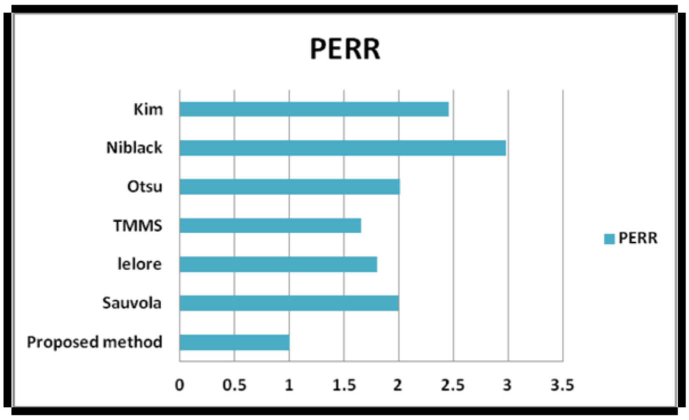

Figure 4.14 Bar graph showing various binarization methods and corresponding PERR value

Pixel Error Rate abbreviated as PERR is defined as $PERR = \frac{Pixelerror}{M\ x\ N}$. It gives the total number of misidentified pixels in the binarization (i.e., total number of those pixels which are of black colour inbinarized image but of white colour in the original image and those which are white in the binarized one but are black in the original image). If $x(i,j)$ represents the value of the pixel situated at i^{th} row and j^{th} column of the original image x and if $y(i,j)$ represents the value of the corresponding pixel (i.e. pixel situated at i^{th} row and j^{th} column) of the binarized image y, the Mean Square Error rate (MSE) is defined as,

$$MSE = \frac{\sum_i \sum_j e(i,j)^2}{M\ x\ N}$$

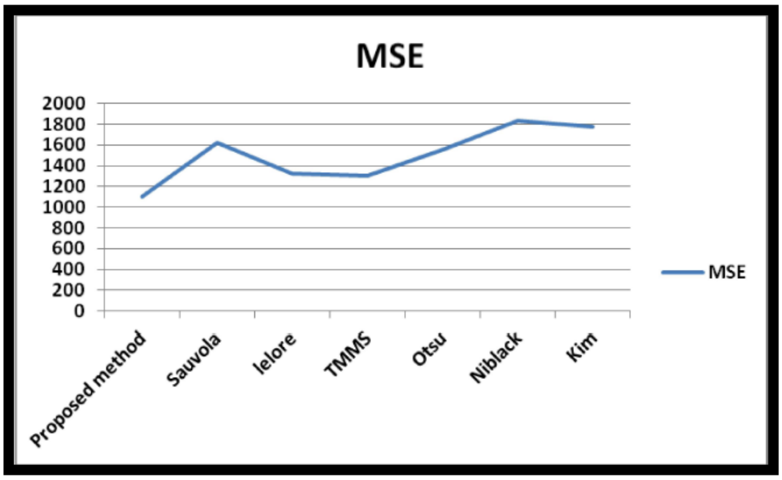

Figure 4.3 Line graph showing various binarization methods and corresponding MSE

As another measurement parameter, we have also used F-measure, which is defined as:-

$$FM = \frac{2x\ Recall\ x\ Precision}{Recall + Precision}$$

From the definition of Pixel Error Rate (PERR), smaller value of the PERR indicates that the algorithm for which PERR is measured is very accurate in terms of binarization results. In the same manner, smaller value of MSE (Mean Square Error) leads to the conclusion that, the quality of binarization produced by the algorithm is of high quality. On the contrary to this, the other third measurement technique, F-measure is quite opposite. Smaller value of F-measure proves superiority of the binarization technique.

Note that, our algorithm gives the maximum F-measure value (of 93) among all other methods listed in the table. It is 5 units higher than one of the most popular binarization method called Sauvola method. Thus, with respect to F —measure our method turns out to be superior to the popular methods listed in the table. The other two measures namely MSE and PERR measure some form of error in the binarization results. Hence, smaller values of these measures for the algorithm prove it better from the others. As shown in the table, our method has the value of l101.021 for MSE and 1.00312 for PERR, which is quite less than other binarization methods.

ISSN: 2094-0343
2326-9865

Table 4.2 Comparison of our algorithm (Proposed method) with popular binarization algorithms

Method	FM	MSE	PERR
Proposed method	93	1101.021	1.00312
Sauvola	88	1622.132	2.00012
lelore	90.1	1321.001	1.80114
TMMS	91.2	1300.564	1.65842
Otsu	89	1551.256	2.01239
Niblack	86.4	1832.534	2.98560
Kim	88.1	1776.723	2.45601

In nutshell, the new binarization method based on our mathematical model produces results better than existing and popular binarization methods. The underlying mathematical model efficiently minimizes the objective function and hence theoretically guarantees significant results. However, the only limitation emerging out of experimental results is high computational time, which needs to be addressed by better programming skills and re-coding of the algorithm.

5 CONCLUSION

In the paper, various graph theoretic models involving graph cuts are studied. Graph cut models for mainly three types of structures viz. uniformly smooth structure, segment wise smooth structure and universally constant structure are studied in the paper. Optimization of variety of objective functions using graph cuts in different move spaces is studied in detail in the paper. The study concludes that, an objective function can be minimized using graph cuts provided it is FNO- optimizable. Characterizations of two classes of FNO- optimizable functions (O^2 and O^3) are given and many mathematical results in this regard are proved in the paper. These characterizations contribute in easily identifying the image processing problems which can be addressed through graph cuts notion. However, further exploration of the concepts studied/ defined in the paper is required to strengthen the designed mathematical framework. Due to programming limitation on our part, we could not construct a computationally time-effective computer program encoding the graph-cuts model. There is a good scope of improvement in the implementation of graph theoretic models designed in the present work on appropriate programming platform.

REFERENCES

1. A. Ben-Israel and T. N. Greville. Generalized Inverses. Springer-Verlag, 2003.
2. A. Bjork. Numerical Methods for Least Squares Problem. SIAM, 1996.

3. A. Blake and A. Zisserman. Visual Reconstruction. MIT Press, 1987.
4. A. Goldberg and R. Tarjan. A new approach to the maximum flow problem. Journal of the Association for Computing Machinery, 35(4):921 940, October 1988.
5. A. Laurentini. The visual hull concept for silhouette-based image understanding. IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(2):150 162, February 1994.
6. A. Simon, J.C. Pret, and A.P. Johnson, A fast algorithm for bottom-up layout analysis, IEEE Trans Pattern Anal Mach Intell 19 (1997), 273 277.
7. A. V. Goldberg and R. E. Tarjan. A new approach to the maximum- flow problem J. ACM, 35(4):921 940, 1988.
8. A.K. Jain and S. Bhattacharjee, Text segmentation using Gabor Filters for automatic document processing, Mach Vis Appl 5 (1992), 169 184.
9. A.K. Jain and Y. Zhong, Page segmentation using texture analysis, Pattern Recognition 29 (1996), 743 770
10. Alexander Schrijver. A combinatorial algorithm minimizing sub modular functions in strongly polynomial time. Journal of Combinatorial Theory, B80:346 355, 2000.
11. B. Gold lucke and M.A. Magnor. Joing 3D-reconstruction and background separation in multiple views using graph cuts. In IEEE Conference on Computer Vision and Pattern Recognition, pages 683 688, 2003.
12. B. K. P. Horn and B. Schunk. Determining optical flow. Artificial Intelligence, 17:185 203, 1981.
13. B. Thirion, B. Bascle, V. Ramesh, and N. Navab. Fusion of color, shading and boundary information for factory pipe segmentation. In CVPR, pages 349 356. IEEE, 2000.
14. B. V. Cherkassky and A. V. Goldberg. On implementing push-relabel method for the maximum flow problem. Algorithmic a, 19:390 410, 1997.
15. B. Wang, X.-F. Li, F. Liu, and F.-Q. Hu, 2005. Colour text image binarization based on binary texture analysis, Pattern Recogn Lett 26 (2005), 1650 1657.
16. Boris Zalesky. Fast algorithms of bayesian segmentation of images. arXiv, math.PR/0206184, June 2002.
17. C. A. Cocosco, V. Kollokian, R. K. s. Kwan, G. B. Pike, and A. C. Evans. Brain web: Online interface to a 3d mri simulated brain database. NeuroImage, 5:425, 1997.
18. C. Buehler, S. J. Gortler, M. Cohen, and L. Mc Mmillan. Minimal surfaces for stereo vision. In European Conference on Computer Vision, volume 3, pages 885 899, 2002.
19. C. Strouthopoulos and N. Papamarkos, Multi-thresholding of mixed type documents, Eng Appl Artif Intell 13 (2000), 323 343.
20. C. Strouthopoulos, N. Papamarkos, and A. Atsalakis, Text extraction in complex colour documents, Pattern Recognition 35 (2002), 1743 1758.
21. C. Thillou and B. Gosselin, Colour binarization for complex camera-based images, Color imaging X: Processing, hardcopy, and applications, Proceedings of the Electronic Imaging Conference of the International Society for Optical Imaging (5667), 2005, pp. 301 308.
22. C. Wolf, J. Jolion, and F. Chassaing. Text localization, enhancement and binarization in

multimedia documents. ICPR, 4:1037 1040, 2002.

23. C. Wolf, J. Jolion, and F. Chassaing. Text localization, enhancement and binarization in multimedia documents. ICPR, 4:1037 1040, 2002.
24. D. Comaniciu and P. Meer, Mean shift: A robust approach toward feature space analysis, IEEE Trans Pattern Anal Mach Intell 24 (2002), 603 619.
25. D. Doermann, J. Liang, and H. Li. Progress in camera-based document image analysis. ICDAR, 1:606 615, 2003.
26. D. Doermann, J. Liang, and H. Li. Progress in camera-based document image analysis. ICDAR, 1:606 615, 2003.
27. D. Greig, B. Porteous, and A. Seheult. Exact maximum a posterior estimation for binary images. Journal of the Royal Statistical Society, Series B, 51(2):271 279, 1989.
28. D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum a posteriori estimation for binary images. Journal of the Royal Statistical Society. Series B (Methodological), 51(2):271 279, 1989.
29. D. Marr and T.A. Poggio. Cooperative computation of stereo disparity. Science, 194(4262):283 287, October 1976.
30. D. R. Karger, P. Klein, C. Stein, M. Thorup, and N. E. Young. Rounding algorithms for a geometric embedding of minimum multi-way cut in STOC '99: proceeding of the thirty first annual ACM symposium on Theory of computing, pages 668 678, New York, NY, USA, 1999. ACM.
31. D. Schlesinger and B. Flach. Transforming an arbitrary min-sum problem into a binary one. In Technical Report TUD-FI06-01, Dresden University of Technology, 2006.
32. D. Z. Chen and X. Wu. Efficient algorithm for k-terminal cuts on planar graphs. In ISAAC Proceedings of the 12th International Symposium on Algorithms and Computation, pages 332 344, London, UK, 2001. Springer-Verlag.
33. Dan Snow, Paul Viola, and Ramin Zabih. Exact voxel occupancy with graph cuts. In IEEE Conference on Computer Vision and Pattern Recognition, pages 345 352, 2000.
34. Guilherme B. Ribeiro, "Graph Signal Processing in a Nutshell", Journal of Communication and Information Systems, Vol. 33, No.1, 2018.
35. Antonio Ortega, Pascal Frossard, Jelena Kovacevic, Jose M. F. Moura, Pierre Vandergheynst, "Graph Signal Processing: Overview, Challenges and Applications", arXiv:1712.00468v2 [eess.SP] 26 Mar 2018.
36. Sergio Barbarossa, Stefania Sardellitti, "Topological Signal Processing over Simplicial Complexes", arXiv:1907.11577v2 [eess.SP] 13 Mar 2020.
37. Markus Puschel, Chris Wendler, "Discrete Signal Processing with Set Functions", arXiv:2001.10290v2 [cs.IT] 22 Oct 2020.
38. Luana Ruiz, Luiz F. O. Chamon and Alejandro Ribeiro, "Graphon Signal Processing", arXiv:2003.05030v5 [eess.SP] 19 Aug 2021.
39. Raksha Ramakrishna and Anna Scaglione, "Grid-Graph Signal Processing (Grid-GSP): A Graph Signal Processing Framework for the Power Grid", arXiv:2103.06068v1 [eess.SP] 10 Mar 2021.

40. Ya Tu and Yun Lin, “Deep Neural Network Compression Technique Towards Efficient Digital Signal Modulation Recognition in Edge Device”, IEEE ACCESS, Volume 7, 2019.
41. Rachid Boukrab and Alba Pagès-Zamora, “Random-Walk Laplacian for Frequency Analysis in Periodic Graphs”, Sensors 2021, 21, 1275. https://doi.org/10.3390/s21041275.