Split-and-Merge Segmentation of Biomedical Images using Graph Wedgelet Decompositions

Wolfgang Erb[1][0000−0003−3541−5401]

University of Padova - Department of Mathematics "Tullio Levi-Civita"
Via Trieste 63, 35121 Padova, Italy
wolfgang.erb@unipd.it
https://www.math.unipd.it/∼erb/

Abstract. Graph wedgelets are a novel tool for the fast decomposition of images in geometrically meaningful, wedge-shaped, subregions. In this work, we study the usage of graph wedgelets as a promising splitting method in a split-and-merge segmentation scheme for images. We combine adaptive wedgelet splits of images with a simple and classical merging strategy for subregions, and obtain in this way an efficient and robust segmentation of relevant subdomains, that can be used in the segmentation of biomedical images obtained by modalities as, for instance, Magnetic Resonance Imaging.

Keywords: Graph Wedgelets · Geometric Wavelets on Graphs · Split-and-merge Segmentation · Biomedical Imaging

1 Introduction

Combined split-and-merge schemes for image segmentation have been introduced by Horowitz & Pavlidis [12] in the seventies of the last century. The principal idea of these region-based segmentation schemes is to recursively partition an image into small homogeneous parts and then to merge these small building blocks into larger homogeneous segments of the image. The merging step of these schemes is usually more cost intensive since the computational load required for the comparison and the combination of subregions is high. Therefore, a key aspect and at the same time also the challenge of split-and-merge strategies is to implement a splitting scheme that decomposes an image efficiently into a relatively small but geometrically significant number of subblocks, as for instance discussed in Correa-Tome & Sanchez-Yanez [5] and Salembier & Garrido [14].

In this work, we explore the usage of graph wedgelets, as introduced in [9], for the initial geometric partitioning of images within a split-and-merge segmentation framework. In this graph-based approach, an image is represented as a discrete graph that is then divided into a set of geometrically significant subgraphs. These partitions can be encoded easily using a binary wedge partitioning (BWP) tree and efficiently stored as a corresponding sequence of center nodes. The primary advantages of such a discrete splitting strategy lie in its computational efficiency, its simplicity in terms of coding, as well as its adaptability to diverse domains and imaging scenarios.

In the following sections, we will recapitulate the fundamental properties of graph wedgelets for image decomposition and show how this wedgelet-based splitting can be effectively combined with a classical merging procedure to yield a final segmentation of images. We will further analyze the computational complexity and storage requirements of the proposed combined scheme. Finally, we illustrate the practical utility of this approach by applying it to the segmentation of white matter or the extraction of tumor areas in biomedical images obtained from Magnetic Resonance Imaging (MRI).

2 Methodology for image splitting and merging

2.1 The splitting procedure based on graph wedgelets

To split the image into geometrically relevant subregions, we use a graph wedgelet decomposition as introduced in [9]. Graph wedgelets interpret the image as a graph signal and efficiently approximate it with piecewise constant functions on partitions that are encoded in a binary graph partitioning tree and constructed upon hierarchical adaptive wedge splits. This structure allows to store and process the image partitions cost-efficiently by a sequence of graph nodes that encode the entire tree. In [9], tests showed that with identical partition sizes graph wedgelets, similarly to their continuous analogs [8, 10] or to binary space partitioning trees [13] lead to a higher adaptivity and, thus, to an improved approximation quality compared to quadtree decompositions or hierarchical Haar wavelet decompositions. A key advantage of graph wedgelets is their flexibility. The graph structure can be easily adapted to different imaging scenarios. This allows the same approach to be applied not only for rectangular images, but also for the decomposition of more complex data such as 3D scans of magnetic particles [2], vector fields on simplicial grids [1], or distorted images captured by hyper-hemispherical cameras [15]. As graph wedgelets rely on a discrete imaging model, they contrast with continuous approaches that often offer stronger theoretical guarantees, see [4, 6, 7]. In the following, we provide a brief review of graph wedgelets, the details and further references can be found in [9].

Modeling images as graphs. In a graph-based approach to image processing, we interpret an image as a graph $G = (V, E, \mathbf{A}, \mathrm{d})$ in which the single pixels of the image are identified as nodes $V = \{\mathrm{v}_1, \ldots, \mathrm{v}_n\}$, and neighboring pixels v_i and $\mathrm{v}_{i'}$ are connected through edges $e_{i,i'} = (\mathrm{v}_i, \mathrm{v}_{i'})$. We assume that the edges are undirected and denote the set of all edges by E. The symmetric matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is referred to as adjacency matrix, and contains as entries $\mathbf{A}_{i,i'}$ the connection weights of the edges $e_{i,i'} \in E$. We further require a metric distance d on the vertex set V. As standard distance d on V we use the graph geodesic distance, i.e., the length of the shortest path connecting two graph vertices. We will further assume that G is connected, i.e., that the distance between two nodes is always finite.

This graph-based approach is very versatile and allows the usage of different neighborhood models for the pixels in the image. If only the closest neighbors in the image grid get connected to a pixel, this leads to a graph with a sparse adjacency matrix $\mathbf{A}$, yielding fast graph-based algorithms if image processing tasks have to be performed locally in subregions. On the other hand, sometimes it is more useful to model an image as a complete graph in which the edge weights are linked to the inverse of the euclidean distance between two pixels.

In this work, we are interested in the decompositions of images, i.e. of functions $f : V \to \mathbb{R}$ representing images on the vertex set V of the graph G. The functions f on the discrete set V can be also interpreted as vectors $f = [f(\mathrm{v}_1), \ldots, f(\mathrm{v}_n)]^\intercal \in \mathbb{R}^n$ and are referred to as *graph signals*.

Wedge splits. Graph wedgelets rely on an adaptive recursive binary partitioning of the vertex set V in which geometric information of the signal f is included. For this, a binary partitioning tree is generated which depends on the topology of the graph G as well as on the signal f. As elementary splittings of a vertex set, we consider wedge splits.

We call a dyadic partition $\{V_\mathrm{q}, V_{\mathrm{q}'}\}$ of the vertex set V a *wedge split* of V if there exists a pair of nodes $\{\mathrm{q}, \mathrm{q}'\}$ in V such that V_q and $V_{\mathrm{q}'}$ have the form

$$V_\mathrm{q} = \{\mathrm{v} \in V \mid \mathrm{d}(\mathrm{v}, \mathrm{q}) \leq \mathrm{d}(\mathrm{v}, \mathrm{q}')\}, \quad V_{\mathrm{q}'} = \{\mathrm{v} \in V \mid \mathrm{d}(\mathrm{v}, \mathrm{q}) > \mathrm{d}(\mathrm{v}, \mathrm{q}')\}.$$

The center nodes q and q′ uniquely determine the wedge split $\{V_\mathrm{q}, V_{\mathrm{q}'}\}$ of V. This allows to code the splittings very efficiently in terms of their center nodes. Further, if the metric d is the graph geodesic distance and V is connected, then also V_q and $V_{\mathrm{q}'}$ are connected subsets of the graph G. A recursive wedge splitting leads to a binary partitioning tree of the graph.

Binary wedge partitioning (BWP) trees. A BWP tree $\mathcal{T}_Q$ of the graph G with respect to the ordered set $Q = \{\mathrm{q}_1, \ldots, \mathrm{q}_M\} \subset V$ of centers is a binary tree consisting of subsets of the vertex set V that can be ordered recursively in partitions $\mathcal{P}^{(m)}$ of V according to the following steps:

1. The root of the tree $\mathcal{T}_Q$ is the entire set V forming the trivial partition $\mathcal{P}^{(1)} = \{V_{\mathrm{q}_1}^{(1)}\} = \{V\}$. The root is linked to the first node q_1 of Q.
2. Let $\mathcal{P}^{(m)} = \{V_{\mathrm{q}_1}^{(m)}, \ldots, V_{\mathrm{q}_m}^{(m)}\}$ be a partition of V in $\mathcal{T}_Q$ linked to the nodes $\mathrm{q}_i \in V_{\mathrm{q}_i}^{(m)}$, $i \in \{1, \ldots, m\}$, $m < M$. Then, the new node q_{m+1} contained in one subset $V_{\mathrm{q}_j}^{(m)}$ together with the center node q_j define a new wedge split of $V_{\mathrm{q}_j}^{(m)}$ resulting in a new partition $\mathcal{P}^{(m+1)}$ of V into $m+1$ disjoint sets.

The ordered set Q of graph vertices uniquely determines the BWP tree $\mathcal{T}_Q$. This allows to encode the entire BWP tree compactly in terms of the M elements of the set Q. A BWP tree $\mathcal{T}_Q$ contains $2M-1$ subdomains of V, 1 root and $2M-2$ children regions. The M leaves of the tree $\mathcal{T}_Q$ correspond to the sets of the final partition $\mathcal{P}^{(M)} = \{V_{\mathrm{q}_1}^{(M)}, \ldots, V_{\mathrm{q}_M}^{(M)}\}$ of V.

Graph wedgelets. The characteristic functions

$$\omega_{\mathrm{q}_i}^{(m)}(\mathrm{v}) = \chi_{V_{\mathrm{q}_i}^{(m)}}(\mathrm{v}), \quad i \in \{1, \dots m\},\ m \in \{1, \dots, M\},$$

of the sets $V_{\mathrm{q}_i}^{(m)}$ in a BWP tree $\mathcal{T}_Q$ are referred to as *graph wedgelets*. The wedgelets $\{\omega_{\mathrm{q}_i}^{(m)} : i \in \{1, \dots, m\}\}$ form an orthogonal basis for the piecewise constant functions on the partition $\mathcal{P}^{(m)}$.

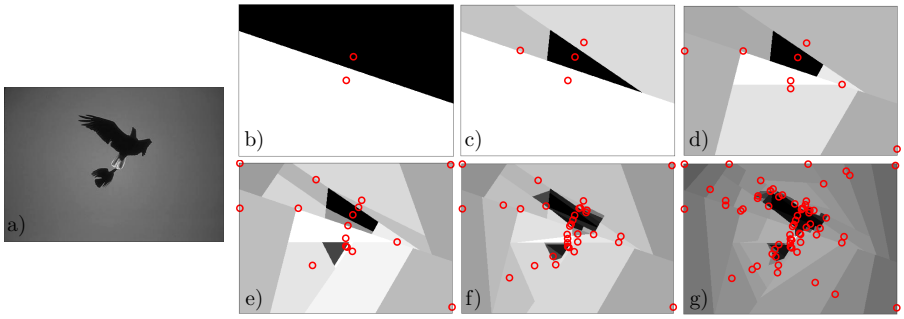

Fig. 1. Adaptive BWP splitting of a 481×321 gray-scale image using Algorithm 1; a) The original image; b)c)d)e)f)g) Graph wedgelet decomposition of the image using 2, 4, 8, 16, 32 and 64 wedgelet domains, respectively. The center nodes Q of the BWP tree are marked with a red circle.

Adaptive greedy calculation of BWP trees. In order to determine the BWP tree $\mathcal{T}_Q$, at each level m one of the subdomains $V_{\mathrm{q}_j}^{(m)}$, $j \in \{1, \dots, m\}$, has to be chosen and a respective new node $\mathrm{q}_{m+1} \in V_{\mathrm{q}_j}^{(m)}$ selected. In order to approximate a signal f best possible, these selections can be performed in an f-adaptive way. We consider the following two-step iterative greedy procedure for the generation of the BWP trees. A simple example of this procedure applied to a gray-scale image can be found in Fig. 1. The entire wedgelet decomposition scheme to generate a BWP tree is summarized in Algorithm 1.

Step 1: Selection of the subdomain: at level m, we consider the mean values

$$\bar{f}_{V_{\mathrm{q}_i}^{(m)}} = \frac{1}{|V_{\mathrm{q}_i}^{(m)}|} \sum_{\mathrm{v} \in V_{\mathrm{q}_i}^{(m)}} f(\mathrm{v})$$

of the function f over the subsets $V_{\mathrm{q}_i}^{(m)}$. Then, the domain $V_{\mathrm{q}_j}^{(m)}$ with the maximal $\mathcal{L}^2$-error

$$j = \operatorname*{argmax}_{i \in \{1, \dots, m\}} \| f - \bar{f}_{V_{\mathrm{q}_i}^{(m)}} \|_{\mathcal{L}^2(V_{\mathrm{q}_i}^{(m)})}, \tag{1}$$

is selected. Inside this domain, the next wedge split is defined in terms of the center node q_j and an additional node q_{m+1}.

Step 2: Adaptive selection of the second node q_{m+1}**.** The second node q_{m+1} is chosen according to an adaptive rule such that the resulting split provides a best possible approximation of the function f in terms of piecewise constant components. If $\{V^{(m)+}_{(\mathrm{q}_j,\mathrm{q})}, V^{(m)-}_{(\mathrm{q}_j,\mathrm{q})}\}$ denotes the partition of $V^{(m)}_{\mathrm{q}_j}$ according to the wedge split given by the node q_j and a second node q, we determine q_{m+1} such that the quantity

$$\|f - \bar{f}_{V^{(m)+}_{(\mathrm{q}_j,\mathrm{q})}}\|^2_{\mathcal{L}^2(V^{(m)+}_{(\mathrm{q}_j,\mathrm{q})})} + \|f - \bar{f}_{V^{(m)-}_{(\mathrm{q}_j,\mathrm{q})}}\|^2_{\mathcal{L}^2(V^{(m)-}_{(\mathrm{q}_j,\mathrm{q})})} \tag{2}$$

is minimized over all $\mathrm{q} \in S \subset V^{(m)}_{\mathrm{q}_j}$. As a set S of admissible nodes, one can for instance take $S = V^{(m)}_{\mathrm{q}_j} \setminus \{\mathrm{q}_j\}$. It is however generally advisable to reduce the computational complexity of the optimization problem (2) by selecting a relatively small set S. One way to do this is by picking $|S|$ nodes uniformly at random from the domain $V^{(m)}_{\mathrm{q}_j} \setminus \{\mathrm{q}_j\}$.

Algorithm 1: Wedgelet decomposition of the image

Input: A graph signal f, a starting node $\mathrm{q}_1 \in V$, the starting partition $\mathcal{P}^{(1)} = \{V\} = \{V^{(1)}_{\mathrm{q}_1}\}$ and a final partition size $M \geq 1$.

for $m = 2$ *to* M **do**

1) **Greedy selection of subset:** calculate j according to the rule (1) as

$$j = \underset{i \in \{1,\ldots,m-1\}}{\arg\max} \left\| f - \bar{f}_{V^{(m-1)}_{\mathrm{q}_i}} \right\|_{\mathcal{L}^2(V^{(m-1)}_{\mathrm{q}_i})};$$

2) **Determine new node** q_m such that the squared $\mathcal{L}^2$-error term (2) is minimized over all nodes in S and add it to the node set Q;

3) Generate **new partition** $\mathcal{P}^{(m)}$ from the partition $\mathcal{P}^{(m-1)}$ by a wedge split of the subset $V^{(m-1)}_{\mathrm{q}_j}$ into the children sets $V^{(m-1)+}_{(\mathrm{q}_j,\mathrm{q}_m)}$ and $V^{(m-1)-}_{(\mathrm{q}_j,\mathrm{q}_m)}$;

4) Compute **mean values** $\bar{f}_{V^{(m)}_{\mathrm{q}_i}}$, $i \in \{1,\ldots,m\}$, for the new partition $\mathcal{P}^{(m)}$ by an update from $\mathcal{P}^{(m-1)}$.

Output: node set $Q = \{\mathrm{q}_1,\ldots,\mathrm{q}_M\}$, mean values $\{\bar{f}_{V^{(M)}_{\mathrm{q}_1}},\ldots,\bar{f}_{V^{(M)}_{\mathrm{q}_M}}\}$.

Acceleration possibilities. For very large images, the adaptive strategy described in 1) and 2) might be too costly from a computational point of view, even if the size $|S|$ of the selected nodes in (2) is relatively small. To reduce the computational expenses, one can additionally split the image in a preliminary step in J subblocks. This can be done by hand by selecting proper subblocks. Alternatively, a clustering algorithm, as for instance k-center clustering [11], can be applied in order to extract subregions of the image, similarly as described for partition of unity methods on graphs in [3]. After this preliminary splitting, the described greedy method in 1) and 2) can be applied separately to each cluster, providing also a direct way to parallelize the computations.

From an implementation point of view, this preliminary splitting into J subblocks results in J disjoint BWP trees instead of a single tree, and the respective selected center nodes generally differ from the ones generated by the single BWP scheme. This modification will therefore also have an impact on the final splitting of the image, as well as for the subsequent merging procedure.

2.2 The merging procedure

Algorithm 2: Merging part of the split-and-merge segmentation

Input: A partition of the vertex set V in M regions $\mathcal{R}^{(M)} = \{R_1, \dots R_M\}$ with $R_i = V_{\mathrm{q}_i}^{(M)}$ and initial values $f_{R_i} = \bar{f}_{V_{\mathrm{q}_i}^{(M)}}$ for $i \in \{1, \dots, M\}$.

for $m = M$ *to* $L+1$ **do**

1) **Selection of regions:** determine the regions R_i, R_j in $\mathcal{R}^{(m)}$ such that

$$O(R_i, R_j) = \min(|R_i|, |R_j|)(f_{R_i} - f_{R_j})^2$$

gets minimal.

2) **Merge** regions R_i and R_j to $R_i \cup R_j$ and calculate **high median**

$$f_{R_i \cup R_j} = \begin{cases} f_{R_i} & \text{if } |R_i| > |R_j|, \\ f_{R_j} & \text{if } |R_i| < |R_j|, \\ \max\{f_{R_i}, f_{R_j}\} & \text{if } |R_i| = |R_j|. \end{cases}$$

3) **Update** $\mathcal{R}^{(m-1)} = \mathcal{R}^{(m)} \setminus \{R_i, R_j\} \cup \{R_i \cup R_j\}$ and the values $f_{R_1}, \dots, f_{R_{m-1}}$ for the sets in the new partition $\mathcal{R}^{(m-1)}$.

Output: Partition $\mathcal{R}^{(L)} = \{R_1, \dots, R_L\}$ of V and values $f_{R_1}, \dots, f_{R_L}$.

Once the image is decomposed into M wedgelet domains $\{V_{\mathrm{q}_1}^{(M)}, \dots, V_{\mathrm{q}_M}^{(M)}\}$ with a sufficient homogeneity which can be measured in terms of the $\mathcal{L}^2$-errors in (1), a merging scheme needs to be applied to the subdomains in order to obtain a segmentation of the image. For this, we will consider as starting information the mean values $\left\{\bar{f}_{V_{\mathrm{q}_1}^{(M)}}, \dots, \bar{f}_{V_{\mathrm{q}_M}^{(M)}}\right\}$ of the function f on the M subdomains and proceed similarly as described in Salembier & Garrido [14] to generate with a bottom-up strategy a second binary partitioning tree for the merging part. For this merging partitioning tree a *merging order* and a *region model* is required. As a model for the image value $f_{R_1 \cup R_2}$ on the union $R_1 \cup R_2$ of two regions R_1 and R_2 we use the upper median of the function values f_{R_1} and f_{R_2} on the regions R_1 and R_2. The similarity between two regions R_1 and R_2 is measured by the quantity

$$O(R_1, R_2) = \min(|R_1|, |R_2|)(f_{R_1} - f_{R_2})^2.$$

According to this measure, the merging starts with those regions where the quantity $O(R_1, R_2)$ is minimal. The merging scheme is completed, and the image segmentation terminates if a selected partition size L of composed subregions is reached. This merging procedure is summarized in Algorithm 2.

Computational complexity and storage costs As outlined out in [9], the computational complexity for the calculation of the adaptive BWP tree given in Section 2.1 can be bounded by $\mathcal{O}(M|S|n)$. On the other hand, the simple merging scheme in Section 2.2 requires $\mathcal{O}(M^2)$ operations for the calculation of the merging tree. For this, the complete split-and-merge segmentation remains cost-efficient as long as the number M of calculated wedgelet domains remains of moderate size.

It is further shown in [9] that the storage costs of the adaptive BWP tree can be bounded by $\lceil \log_2(n) \rceil M/n$ bits per pixel. In order to store the segmentation of the image in L subregions, only additional $\lceil \log_2(L) \rceil M/n$ bits per pixel have to be invested. If, on top, also the region values $f_{R_1}, \ldots, f_{R_L}$ are stored, further $\lceil \log_2(K) \rceil L/n$ bits per pixel are required if the values f_{R_i} are quantized in terms of K different values.

3 Results and Discussions

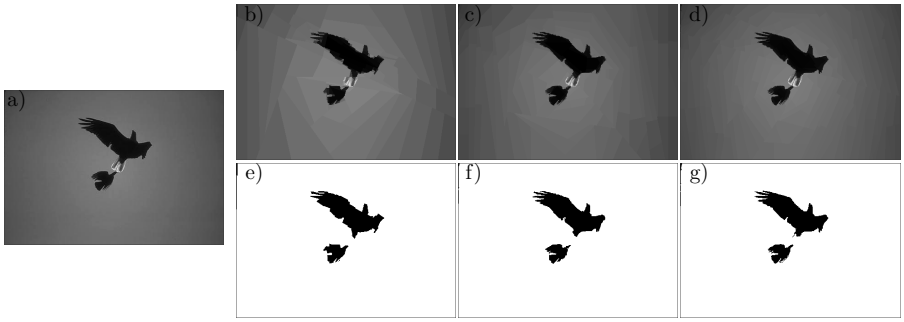

Fig. 2. Split-and-merge segmentation of a gray-scale image a) original image with 481×321 pixels; b)c)d) BWP decomposition of the image using 200, 500 and 1000 domains; e)f)g) Image segmentation based on the application of Algorithm 2 to the BWP decompositions in b)c)d). Segmentations with 2 components are created.

We test our split-and-merge scheme on two gray-scale images and two MRI data sets. The 481×321 gray-scale images in Fig. 2 a) and Fig 3 a) are used as simple test sets to obtain an initial assessment of the method and a rough indication on how the number M of domains and the choice of the set S in the initial BWP splitting influence the final segmentation of the image. For this, we generate in Fig. 2 b)c)d) a BWP decomposition of the image using 200, 500 and 1000 wedgelet domains, and subsequently apply the merging procedure in Algorithm 2 to obtain respective segmentations of the image with $L = 2$ components. Further, in Fig. 3, Algorithm 1 is applied with the following parameter configuration. The subfigures b)c)d) illustrate the wedgelet decomposition of the image into $M = 1000$ regions, where b) the sets S consist of a single node located

farthest away from the center node q_j, c) the sets S consist of 1000 randomly selected nodes, and d) the selection sets S are the same as in c) but a preliminary partitioning of the image in $J = 25$ subblocks is used. In the subfigures e)f)g) of Fig. 3 the respective segmentation provided by Algorithm 2 is shown.

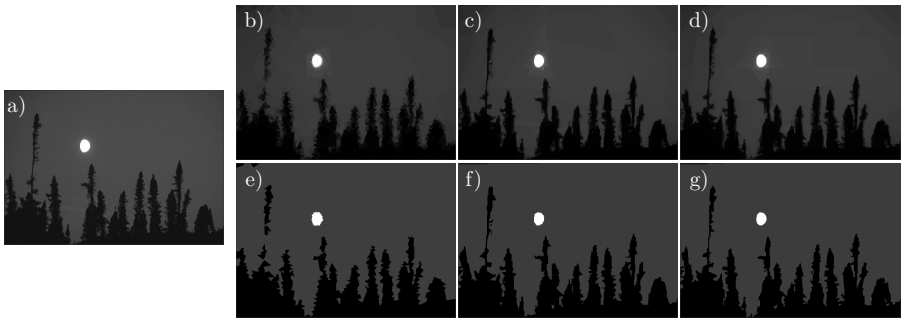

Fig. 3. Split-and-merge segmentation of a gray-scale image a) original image with 481×321 pixels; b)c)d) BWP decomposition of the image using $M = 1000$ domains and b) sets S consisting of a single node, c) sets S with 1000 randomly selected nodes, and d) an additional preliminary partitioning of the image in $J = 25$ blocks; e)f)g) Image segmentation based on the application of Algorithm 2 to the BWP decompositions in b)c)d). Segmentations with 3 components are created.

The third data set is an artificial MRI image of the brain (in Fig. 4 a)) generated by the simulator BrainWeb (http://www.bic.mni.mcgill.ca/brainweb/). Using the presented split-and-merge approach, our goal is to segment the white matter from the rest of the surrounding brain tissue. Using the graph wedgelet algorithm, we split the original MRI image in 2500 subdomains (Fig. 4 b), PSNR 33.19). Then, we apply the merging procedure of Algorithm 2 and obtain a segmentation of the image in terms of $L = 4$ different grayscales (Fig. 4 c)). The segment related to the white matter is afterwards colored in blue (Fig. 4 d)).

In Fig. 5, we additionally compare the wedgelet-based extraction of the white matter (Fig. 5 c)) with the ground truth of the white matter downloaded from BrainWeb (Fig. 5 a)). As the ground truth contains different gray-scales, we extract the domain representing the white matter by a simple thresholding strategy (Fig. 5 b)). The absolute difference between the domains in the subfigures b) and c) is then illustrated in Fig. 5 d).

The forth test image is an MRI scan containing a glioma (in Fig. 6 a)) taken from a dataset on Kaggle (https://www.kaggle.com/sartajbhuvaji/brain-tumor-classification-mri/). The aim of this final experiment is to see whether our approach can segment the glioma from the rest of the image. Again, we conduct a wedgelet decomposition (Fig. 6 b), PSNR 27.68) to split the image in 2500 subregions. After the application of the merging algorithm, we obtain a segmentation of the image in $L = 4$ grayscales (Fig. 6 c)). The segment describing the region with the glioma is colored in blue (Fig. 6 d)).

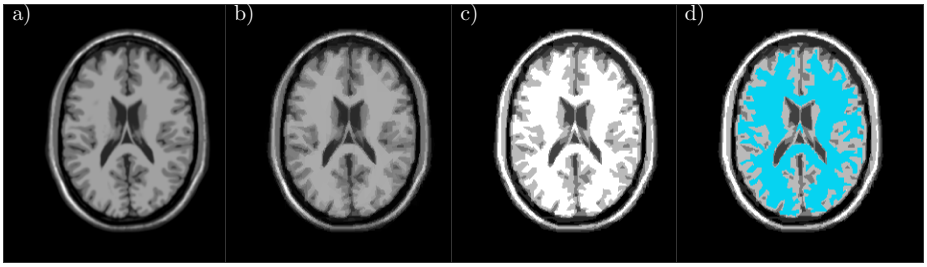

Fig. 4. Wedgelet split-and-merge segmentation for biomedical images: a) Original artificial MRI image; b) Wedgelet BWP split into 2500 regions; c) Segmentation with the presented method; d) Blue colored segmented white matter

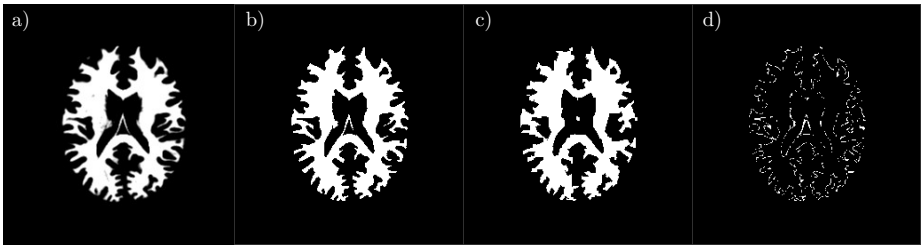

Fig. 5. Comparison of wedgelet split-and-merge segmentation with ground truth of white matter: a) Ground truth of white matter of MRI image in Fig. 4; b) Extraction of segmentation domain from ground truth by thresholding; c) Segmentation of white matter with the presented method; d) Absolute difference between b) and c)

In all four experiments, it is visible that the wedgelet algorithm provides a piecewise constant approximation of the original image and a splitting into a moderate number of geometrically significant wedge-shaped subregions. These preselected subregions can then be easily processed with the given merging method to obtain a simple and fast segmentation of the aimed-at regions. An advantage of the used wedgelet decomposition in the splitting procedure is that the original image is automatically denoised, leading to a more robust final segmentation. On the other hand, it is also visible that some details of the image get lost in those regions where the wedgelet approximation is not accurate enough. This is particularly visible in Fig. 2 where different numbers M of wedgelet splits are compared, and in Fig. 5 where differences at the border of the extracted white matter domain get visible in comparison to the ground truth.

The wedgelet algorithm itself allows to select the number M of adopted wedge splits and the sets S of nodes which are taken into consideration for the Greedy selection of the splits. This allows to control the number of considered subregions and the accuracy of the wedgelet approximation. The role of the number M is, for instance, visualized in Fig. 2, while in Fig. 3 it gets visible that large selection sets S lead to a more accurate partitioning of the domain

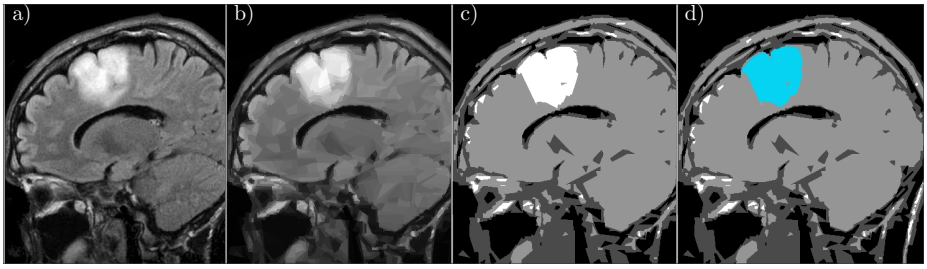

Fig. 6. Wedgelet split-and-merge segmentation for biomedical images: a) Original MRI image with glioma b) Wedgelet split into 2500 regions c) Segmentation with the presented method d) Blue colored segmented glioma.

and a refined segmentation. The usage of a preliminary splitting of the domain in J subblocks has only a relatively small impact on the final segmentation. The selection of these parameters and the trade-off between accuracy and robustness plays an important role also for the computational complexity of the method and needs to be investigated further to guarantee significant but also computationally efficient segmentations of medical images. Furthermore, in this study we used only a very simple model for the merging step. This leaves a lot of potential for further improvements.

4 Conclusion

In this short study, we explored a graph-based image splitting strategy as part of an integrated split-and-merge segmentation framework with particular emphasis on applications in biomedical imaging. The presented splitting method leverages discrete adaptive wedge splits to divide images into a set of geometrically meaningful subregions that can be described by a binary wedge partitioning tree and encoded efficiently by a sequence of graph centers. This graph-based splitting scheme together with a classical merging procedure yields a robust and computationally efficient split-and-merge segmentation scheme for images. The presented numerical experiments demonstrate that the method has strong potential and can be applied practically to segmentation tasks in medical imaging.

Acknowledgments. The author of this article is funded by the Università degli Studi di Padova - Dipartimento di Matematica under the project SID BIRD 2023 entitled "ALISIA - ALgorithms for Immersive Stereoscopic Imaging with Appliciations to the Daedalus camera system", and by the European Union – NextGenerationEU under the National Recovery and Resilience Plan (NRRP), Mission 4 Component 2 Investment 1.1 - Call PRIN 2022 No. 104 of February 2, 2022 of Italian Ministry of University and Research; Project 2022FHCNY3 (subject area: PE - Physical Sciences and Engineering) "Computational mEthods for Medical Imaging (CEMI)". Further support by the Indam research group GNCS (Indam-GNCS project CUP_E53C24001950001), the Italian Research Network on Approximation (RITA), and the Italian Mathematical Union (research group UMI-TAA) is gratefully acknowledged.

References

1. Alonso Rodríguez, A., Bruni Bruno, L., Rapetti, F.: Computing weights for high order Whitney edge elements, Dolomites Res. Notes Approx. **15**, 1-12 (2022).
2. Bringout, G., Erb, W. and Frikel, J.: A new 3D model for Magnetic Particle Imaging using realistic magnetic field topologies for algebraic reconstruction. Inverse Problems **36**, 124002(2020)
3. Cavoretto, R., De Rossi, A., Erb, W.: Partition of Unity Methods for Signal Processing on Graphs. J. Fourier Anal. Appl. **27**, Art. 66. (2021)
4. Cohen, A., Dyn, N., Hecht, F., Mirebeau, J-M.: Adaptive multiresolution analysis based on anistropic triangulations. Math. Comput. **81**(278), 789–810 (2012)
5. Correa-Tome, F.E., Sanchez-Yanez, R.E.: Integral split-and-merge methodology for real-time image segmentation. J. of Electronic Imaging, **24**(1), 013007 (2015)
6. Dekel, S., Leviatan, D.: Adaptive multivariate approximation using binary space partitions and geometric wavelets. SIAM J. Numer. Anal. **43**(2), 707–732 (2005)
7. Demaret, L., Iske, A.: Optimal N-term approximation by linear splines over anisotropic Delaunay triangulations. Math. Comput. **84**(293), 1241–1264 (2015)
8. Donoho, D.L.: Wedgelets: Nearly minimax estimation of edges. Annals of Statistics **27**(3), 859–897 (1999)
9. Erb, W.: Graph Wedgelets: Adaptive Data Compression on Graphs based on Binary Wedge Partitioning Trees and Geometric Wavelets. IEEE Trans. Image Process. **9**, 24–34 (2023)
10. Friedrich, F., Demaret, L., Führ, F., Wicker, K.: Efficient Moment Computation over Polygonal Domains with an Application to Rapid Wedgelet Approximation. SIAM J. Sci. Comput. **29**(2), 842–863 (2007)
11. Gonzalez, T.F.: Clustering to minimize the maximum intercluster distance. Theoretical Computer Science **38**, 293–306 (1985)
12. Horowitz, S., Pavlidis, T.: Picture segmentation by a tree traversal algorithm, Journal of the ACM **23**, 368–388 (1976)
13. Radha, H., Vetterli, M., Leonardi, R.: Image compression using binary space partitioning trees. IEEE Trans. Image Proc. **5**(12), 1610–1624 (1996)
14. Salembier, P., Garrido, L.: Binary Partition Tree as an Efficient Representation for Image Processing, Segmentation, and Information Retrieval. IEEE Trans. Image Process.**9**(4), 561–576 (2000)
15. Simioni, E., Pernechele, C., Erb, W., Marchini, A., Martini, P., Lessio, L., Penasa L., Beghini, M.: A-Central Model for the Geometric Calibration of Hyperhemispherical Lenses. Opt. Express **32**(20), 34777–34795 (2024)