Split-and-Merge Segmentation based on Graph Wedgelets

Wolfgang Erb
University of Padova, Department of Mathematics
wolfgang.erb@unipd.it

***Abstract*—Graph wedgelets are novel adaptive building elements for the sparse decomposition of graph signals in geometrically meaningful, wedge-shaped, subregions. In this work, we study the usage of graph wedgelets as a splitting method in a combined split-and-merge scheme for the segmentation of 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 domains in images.**

I. Introduction

Combined split-and-merge schemes for image segmentation have been introduced by Horowitz and Pavlidis [1] 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 the cost-intensive part as the computational load required for the comparison and the combination of subregions is large. 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, cf. [2] and [3]. In this work, we will shortly illustrate that graph wedgelets as introduced in [4] constitute such an aimed-at tool for the geometric splitting part inside a combined split-and-merge segmentation procedure.

II. Graph Wedgelets: an adaptive tool for the generation of binary image partitionings

To split the image into regions that take into account the underlying geometry of the image we use a graph wedgelet decomposition as introduced in [4]. Graph wedgelets interpret the image as a graph signal and sparsely approximate it with piecewise constant functions on adaptive partitions that are encoded in a binary graph partitioning tree. The hierarchical splitting into nested partitions is provided by so-called wedge splits of the graph nodes. A dyadic partition $\{R_1, R_2\}$ is called a *w*edge split of a set R if there exist two distinct points q_1 and q_2 in R such that R_1 and R_2 have the form

$$\begin{aligned} R_1 &= \{r \in R \mid \mathrm{d}(r, q_1) \leq \mathrm{d}(r, q_2)\}, \quad \text{and} \\ R_2 &= \{r \in R \mid \mathrm{d}(r, q_1) > \mathrm{d}(r, q_2)\}. \end{aligned}$$

Here d denotes a metric distance on R. The usage of these elementary wedge splits in terms of the center nodes q_1 and q_2 allows to store and process the image partitions cost-efficiently by a sequence of nodes q_i that encode the entire wedge partitioning tree. The wedge splits are performed in an adaptive way such that a given signal f on the set R is optimally approximated by piecewise constant functions on the newly generated partition $\{R_1, R_2\}$.

Graph wedgelets can be regarded as a discrete analog of binary space partitioning trees or of continuous wedgelets in the bivariate setting. In [4], it was shown that adaptive graph wedgelets lead to sparse representations of images with piecewise constant functions that outperform, for instance, quadtree decompositions or hierarchical Haar wavelet decompositions.

III. The merging scheme

Once we have calculated a wedgelet partition of the image, a merging scheme is applied to the respective subdomains in order to obtain a segmentation of the image. We proceed similarly as in [3] and generate a second binary partitioning tree for the merging part. For this, a merging order and a region model is required. As a model for the image value on the union $R_1 \cup R_2$ of two regions, we use the medians F_1 and F_2 of the function values on R_1 and R_2. The similarity between R_1 and R_2 is then measured by the quantity

$$O(R_1, R_2) = \min(N_1, N_2)(F_1 - F_2)^2,$$

where N_1 and N_2 denote the number of pixels in the respective subregions. 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 K of composed subregions is reached.

IV. Some experimental results

We test our scheme on two different images from the Berkeley Segmentation Data Set and Benchmarks 500 (BSDS500) [5]. The first test image is a gray-scale image displaying the contours of two birds. Using the presented split-and-merge approach, our goal is to segment the two birds from the background. Using graph wedgelets, we split the image in 200, 500 and 1000 regions (Fig.1 b)c)d), PSNR 34.121773, 37.350934 and 40.305822) and apply the merging algorithm of Section III. In this way, we obtain a segmentation of the image in terms of two grayscales (Fig. 1 e)f)g), respectively).

The second test image is a colored image displaying a church and contains more refined details than the first test image. Again, we conduct a wedgelet decomposition (Fig.2 b)c)d), PSNR 28.189883, 30.976956 and 34.042697) to split the image in 500, 1000, and 2000 regions. After applying the merging methods, we obtain a segmentation of the image in 14 different colors (Fig. 2 e)f)g)).

V. Discussion and Conclusion

In both experiments, it is visible that the wedgelet algorithm gives a piecewise constant approximation of the original image and a splitting into geometrically significant wedge-shaped subregions. These calculated regions can be easily processed with the described merging method to obtain a simple segmentation of the images. An advantage of the wedgelet decomposition in the splitting procedure is an automatic denoising of the image, leading to a more robust final segmentation. On the other hand, we also see that some details of the images get lost in those regions where the wedgelet approximation is not accurate enough. The wedgelet algorithm allows to select the number of adopted wedge splits, and, thus, to control the number of subregions and the accuracy of the image approximation. This trade-off between accuracy and robustness plays an important role also for the computational complexity of the method and needs to be investigated further, see also the discussion in [6]. Furthermore, in this study we used only a very simple model for the merging step. This leaves a lot of potential for further improvement.

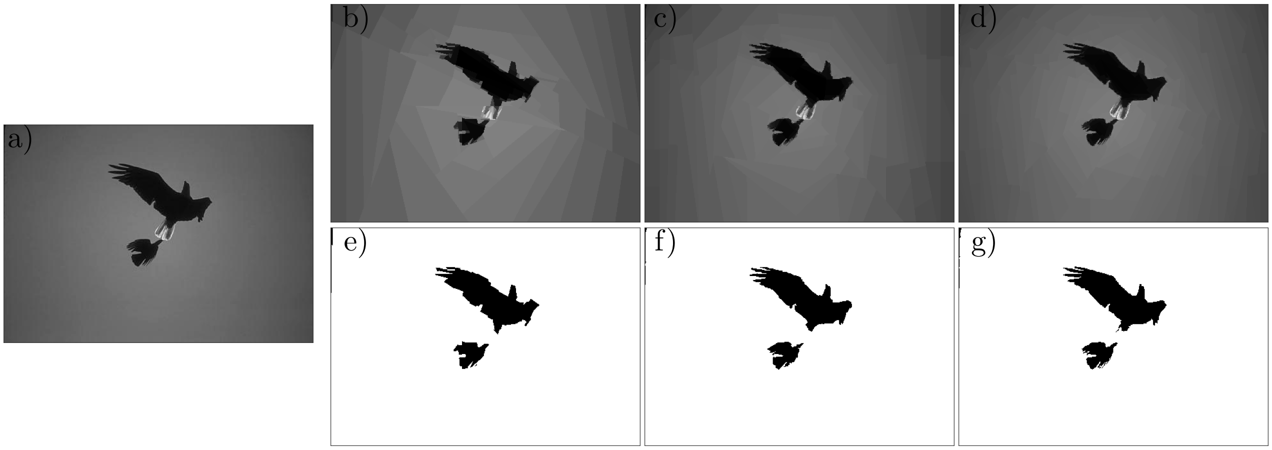

Fig. 1. Split and merge segmentation of gray-scale image: a) original image with 481×321 pixels; b)c)d) Wedgelet approximation with 200, 500 and 1000 graph wedgelets; e)f)g) Segmentation of image by merging the partitions in b)c)d), respectively.

Fig. 2. Split and merge segmentation of colored image: a) original image with 481×321 pixels; b)c)d) Wedgelet approximation with 200, 500 and 1000 graph wedgelets; e)f)g) Segmentation of image by merging the decompositions in b)c)d), respectively.

References

[1] S. Horowitz and T. Pavlidis, *Picture segmentation by a tree traversal algorithm*, Journal of the ACM 23 (1976), pp. 368-388.

[2] F.E. Correa-Tome and R.E. Sanchez-Yanez, *Integral split-and-merge methodology for real-time image segmentation*, J. of Electronic Imaging, Vol. 33, No. 1, 013007, 2015.

[3] P. Salembier and L. Garrido, *Binary Partition Tree as an Efficient Representation for Image Processing, Segmentation, and Information Retrieval*, IEEE Trans Image Process., Vol 9, No. 4, pp. 561-576, 2000.

[4] W. Erb, *Graph Wedgelets: Adaptive Data Compression on Graphs based on Binary Wedge Partitioning Trees and Geometric Wavelets*, arXiv:2110.08843, 2021

[5] P. Arbelaez, M. Maire, C. Fowlkes and J. Malik, *Contour Detection and Hierarchical Image Segmentation*, IEEE TPAMI, Vol. 33, No. 5, pp. 898-916, 2011.

[6] R. Cavoretto, A. De Rossi and W. Erb, *Partition of Unity Methods for Signal Processing on Graphs*, J. Fourier Anal. Appl., Vol. 27, Vol. 33, Art. 66, 2021.