Scale and Edge Detection with Topological Derivatives

Guozhi Dong[1], Markus Grasmair[1,2], Sung Ha Kang[3], and Otmar Scherzer[1,4]

[1] Computational Science Center,
University of Vienna,
Nordbergstrasse 15,
1090 Wien, Austria
{guozhi.dong,markus.grasmair,otmar.scherzer}@univie.ac.at
[2] Catholic University Eichstätt–Ingolstadt,
Ostenstrasse 26,
85072 Eichstätt, Germany
[3] School of Mathematics,
Georgia Institute of Technology,
686 Cherry Street NW
Atlanta, GA 30332-0160, USA
kang@math.gatech.edu
[4] Johann Radon Institute for Computational and
Applied Mathematics (RICAM),
Austrian Academy of Sciences,
Altenbergerstrasse 69, A-4040 Linz, Austria

Abstract. A typical task of image segmentation is to partition a given image into regions of homogeneous property. In this paper we focus on the problem of further detecting scales of discontinuities of the image. The approach uses a recently developed iterative numerical algorithm for minimizing the Mumford-Shah functional which is based on topological derivatives. For the scale selection we use a squared norm of the gradient at edge points. During the iteration progress, the square norm, as a function varied with iteration numbers, provides information about different scales of the discontinuity sets. For realistic image data, the graph of the norm function is regularized by using total variation minimization to provide stable separation. We present the details of the algorithm and document various numerical experiments.

Keywords: Mumford-Shah Functional, Topological Derivatives, Scale Selection, Total Variational Filtering

1 Introduction

One of the most well-studied image segmentation model is the Mumford–Shah functional [16], which is to find the image u which minimizes the following:

$$F(u,K) = \frac{1}{2}\int_{\Omega}(u-f)^2 dx + \frac{\alpha}{2}\int_{\Omega\setminus K}|\nabla u|^2 dx + \beta\mathcal{H}^1(K) \qquad (1.1)$$

over all sets $K \subset \Omega$ and all smooth functions u defined on $\Omega \backslash K$. The first component provides a piecewise smooth approximation of the given image data $f\colon \Omega \to [0, \infty)$. The second component provides the information on the discontinuity set of the image f. Here $\mathcal{H}^s(K)$ denotes the s-dimensional Hausdorff-measure of the set K. We focus on the two-dimensional case $\Omega \subset \mathbb{R}^2$.

There are number of methods proposed to minimize the Mumford-Shah functional. One of the most important approach is by Ambrosio and Tortorelli [1, 4]. The general idea is to approximate the functional by a family of elliptic functionals, where each of them in principle can be minimized with numerical partial differential equation solvers. Related works include the Chan–Vese model [7], where the Mumford–Shah model is simplified to the reconstruction of piecewise constant functions only, and the discontinuity sets are eliminated using an explicit notion of boundary via a level set formulation or using well-potential models (see [13, 21]).

Recently, a numerical algorithm for minimizing the Mumford–Shah functional based on topological derivatives has been developed [12]. The implementation of the algorithm is iterative in nature and selects edges successively according to certain rules. In this paper, we further experimentally analyse these criteria. We show that the algorithm based on topological derivatives can distinguish between edges of different scales and therefore can be used for detecting scales of edges. This is different from [1, 4] where the global approximation of the Mumford–Shah functional is achieved by partial differential equations, which, however, does not allow a selective selection of edges.

The approach of [12] consists of approximating the Mumford–Shah functional by the family of functionals

$$\begin{aligned} J_{\varepsilon,\kappa}(u,K) &= G_{\varepsilon,\kappa}(u,K) + 2\beta\varepsilon m_\varepsilon(K) \\ &= \frac{1}{2}\int_\Omega (u-f)^2 dx + \frac{\alpha}{2}\int_{\Omega\backslash K} |\nabla u|^2\, dx + \kappa\frac{\alpha}{2}\int_{K\cap\Omega} |\nabla u|^2\, dx + 2\beta\varepsilon m_\varepsilon(K)\,, \end{aligned} \tag{1.2}$$

where

$$m_\varepsilon(K) = \inf\{\mathcal{H}^0(Y) : Y \subset \mathbb{R}^2,\ K = \bigcup_{y\in Y} B_\varepsilon(y)\}.$$

The minimization is performed over all $u \in H^1(\Omega)$ and $K \subset \mathbb{R}^2$. It has been shown in [12] that these functionals Γ-converge to F, if $\kappa = o(\varepsilon)$. For fixed ε and κ the approximate minimization of the functional $J_{\varepsilon,\kappa}$ is performed by using a *topological asymptotic analysis* (see [9, 10, 22]). In the context of image processing, topological asymptotic analysis has been recently applied by Auroux et al. [2, 3] and by Muszkieta [17]. In [12], an implementation for minimizing $J_{\varepsilon,\kappa}$ is proposed, and compared to the Ambrosio–Tortorelli approach [1].

The outline of this paper is as follows: In the following section, we recapitulate the algorithm from [12] for approximate minimization of the Mumford–Shah functional. We present a simple example where we can explain the idea of scales of edges in Section 3. Section 4 considers scale detection for realistic image data. In the later cases, total variation regularization of the according scale detection functions over the number of iterations has to be performed to be able to calculate the according edges. We use the taut string algorithm for computing the total variation minimizers.

2 A Topological Algorithm for Edge Detection

We shortly review the algorithm from [12] for detecting edges in an image $f\colon \Omega \to \mathbb{R}$. In this iterative algorithm, the edges are approximated by a collection of balls of small diameter 2ε (in implementations, the diameter is chosen as the pixel distance). In each iteration, we smooth the original image using a diffusivity which is small at the (previously found) edge set and large outside this set. Then we add the new balls where the gradient norm is largest to the edge set. A detailed outline is given in Algorithm 1.

Algorithm 1 Topological algorithm for edge detection.

Let $f \in L^\infty(\Omega)$, α, $\beta > 0$, $\varepsilon > 0$ and $0 < \kappa < 1$ be given. Set $k = 0$ and $K_0 := \emptyset$.

Step 1. Define

$$u_k := \arg\min_u G_{\varepsilon,\kappa}(u, K_k).$$

Step 2. For $i = 1, \dots, m$, find $y_k^{(i)} \in \Omega \setminus K_k$ such that $|\nabla u_k(y)|^2$ is maximal, and replace K_k by $K_k \cup \{B_\epsilon(y_k^{(i)})\}$.

Step 3. If

$$\max_i \frac{\alpha}{2} \pi \frac{1-\kappa}{1+\kappa} |\nabla u_k(y_k^{(i)})|^2 < \frac{\beta}{\varepsilon},$$

stop the iteration; else set $K_{k+1} := K_k$, $u := u_k$, increase k by 1, and go to Step 1.

Result: Approximation of an optimal edge set K and smoothed image u for the Mumford–Shah functional with parameters α and β.

It is shown in [12] that the resulting set K and the smoothed image u can be considered approximations of the minimizer of the Mumford–Shah functional.

Remark 1. The parameters α and β in Algorithm 1 are identical with the parameters in the Mumford–Shah functional. For noisy images, they should be chosen in dependence of the noise level: the larger the parameters are the smoother the filtered results are and the smaller the edge sets are. In the numerical experiments, we used α in the range from 5 to 10 and β between 100 and 200, and the size of the images are 256×256, of which the intensities range from 0 to 255.

In the numerical implementations, the parameter ε is always chosen as half the distance between adjacent pixels. According to [12], the parameter κ should be chosen as $o(\varepsilon)$. In our implementations we have set κ equal to 0.005. In general, the results proved quite robust with respect to the variations of κ, except for the optimization problem $G_{\varepsilon,\kappa}(u, K) \to \min$ in Step 2 of the algorithm, which becomes more difficult to solve as κ decreases.

The original algorithm from [12] uses an update of the function u whenever a ball has been added to K; this corresponds to setting $m = 1$ in Algorithm 1. For larger images, this is obviously not feasible. In this paper, we add multiple balls during each iteration, slightly compromising the accuracy in favour of vastly improved computation times.

3 Detecting Scale of Discontinuities

We present how the iterative construction of the edge set K in Algorithm 1 can be applied to detecting the edges of different *scales*. To highlight the idea, we consider the test image depicted in Figure 1, which consists of different flat regions that appear well separable. For this piecewise constant example the different scales of edges correspond to edges of a certain magnitude.

We first define the following norm to distinguish between different scales of the edges:

$$S(k) := |\nabla u_k(y_k^{(1)})|^2 \,. \tag{3.1}$$

This is the squared norm of the gradient of u_k at the center point of the first ball detected in each iteration, i.e., the largest gradient outside of the edge set at the previous iteration.

Figure 1 shows the results of Algorithm 1 for a fixed set of parameters. The function S shows some discriminative features: there are intervals where the values are approximately constant. This reflects that the edges recovered during these iterations are of a similar magnitude. Moreover, there are clear discontinuities (clear drops in height), which reflects that all the edges of a certain scale have been completely detected.

This change in the edge jump is illustrated in the lower images in Figure 1. They present the current edge indicators K_k at steps k of the iteration where the most significant jumps in S occur. In addition, we have chosen the depicted jumps sufficiently far from each other so that the differences between subsequent edge indicators are not too small. The first jump in S appears after approximately 40 iterations. The corresponding edge indicator K_{40} indicates the upper left edges of the image, which have the largest absolute value of the gradient. The next significant jump of the function S occurs at iteration 74 — here we extract the full shape of the spade; the surface of the clubs is fully recovered at iteration 124 and the diamond comes out last around the 182 iterations with a very slight drop of S as the last obviously detectable scale.

4 Regularization of $S(k)$

The Cards image consists of large piecewise constant regions and the edges betweens these regions are clearly pronounced, and different scales of edges are easily identified. For natural images, the scales of edges are less pronounced, and the function S shows a less regular behaviour. This effect worsens if the data are noisy. We experimented with the *Cameraman* and the *Peppers* image data (see

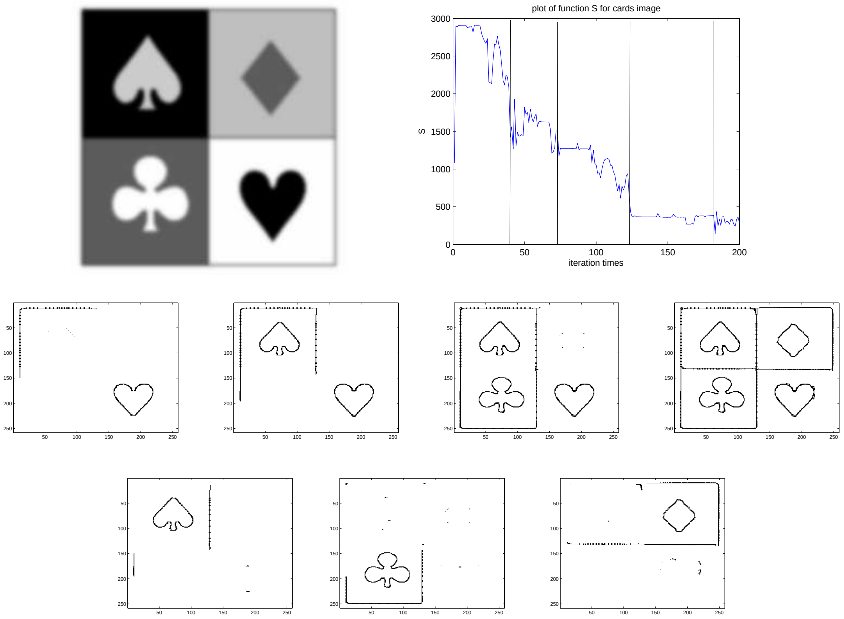

Fig. 1: Cards pictures. We apply Algorithm 1 with parameters $\alpha = 5$, $\beta = 100$, and $m = 20$. *Upper row, left:* The *cards* image. *Upper row, right:* The graph of the function S, one can discern distinct jumps of this function. *Middle row:* The edge set K_k at the iterations k where the most important jumps occur (from left to right: iterations 40, 74, 124, and 182). *Lower row:* The difference between two neighbouring edge sets.

Figure 2), and the functions S do not reveal similar obvious plateaus as for the Cards image.

To better deal with these natural images, we smooth the function S (considered as a function of iterates) by minimizing the discrete total variation, setting

$$\hat{S}_\lambda := \arg\min_R \left(\sum_k (R(k) - S(k))^2 + \lambda |R(k+1) - R(k)| \right).$$

This optimization problem has been studied for a long time in the contexts of one-dimensional signal processing and non-parametric regression (see for instance [8, 15, 23]). There are numerous methods for solving this minimization problem, most of which deal specifically with total variation regularization for image denoising (see for instance [5, 6, 19]). In the one-dimensional case, the most efficient method for total variation minimization is the *taut string algorithm* [8, 11, 18], which can be implemented in the form of a dynamical programming algorithm with linear time and space complexity. A detailed derivation of this method can be found in [11, 20]. The dynamical programming algorithm for the solution is described in [8]. For this algorithm recall that a one-dimensional func-

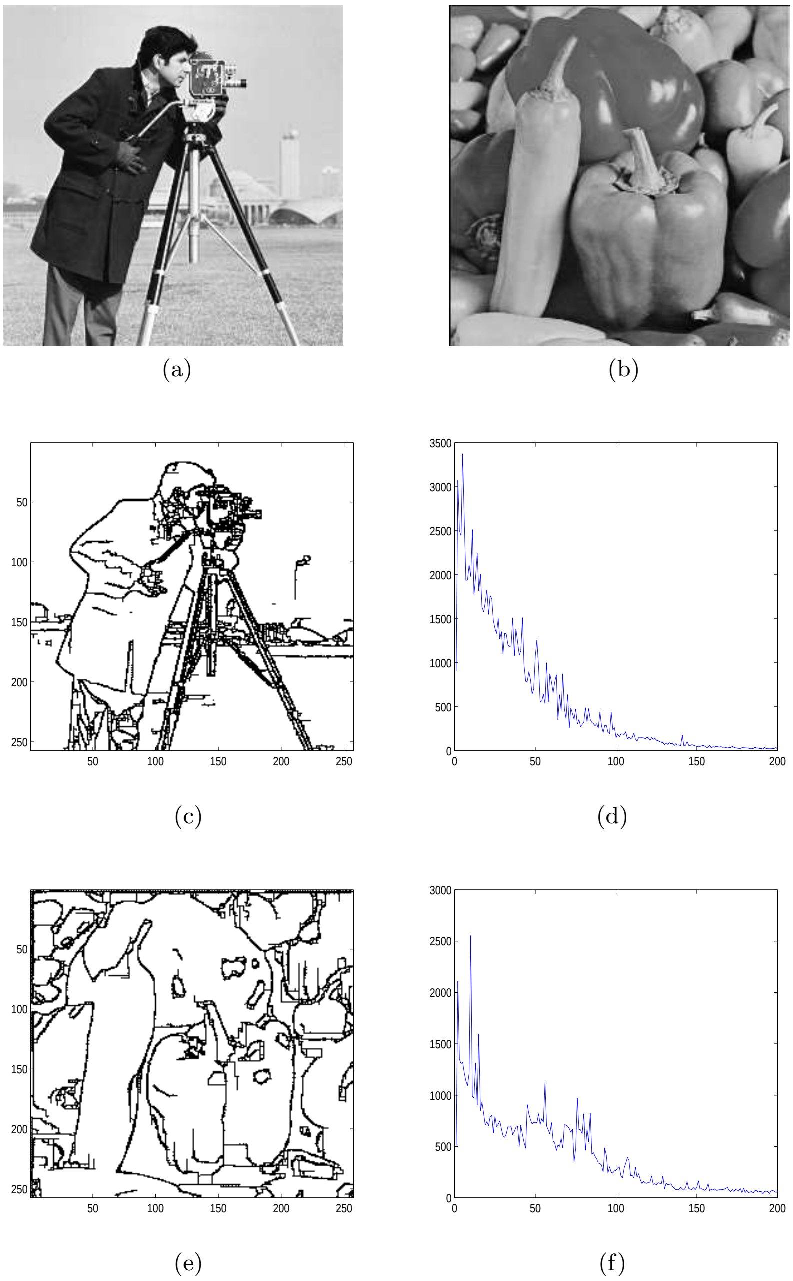

Fig. 2: Edge sets and scale detection function S for the Cameraman and Peppers images. (c) and (e) are the edges of the Cameraman and Peppers images, respectively, detected by our algorithm. In both cases the algorithm has been used with parameters $\alpha = 10$, $\beta = 200$, and $m = 50$. (d) and (f) are the graphs of the functions S for the Cameraman and Peppers images, respectively, where the horizontal axis represents the iteration numbers and the longitudinal axis is the maximum norm of gradients for the center of added balls in that iterations.

tion of bounded variation is continuous outside its jump set, and that a function $U \in W^{1,1}(\Omega)$ is continuous.

Algorithm 2 Taut String Algorithm

Given discrete data $\boldsymbol{u}^\delta = (f_i)$, $i = 1, \dots, s$, and $\lambda > 0$, the taut string algorithm is defined as follows:

Step 1. Let $U_0^\delta = 0$ and $U_i^\delta = \frac{1}{s} \sum_{j=1}^{i} f_j$, $i = 1, \dots, s$. We denote by $U^\delta(x)$ the linear spline with nodal points $x_i = i/s$, $i = 0, \dots, s$, and function values U_i^δ at x_i.

Step 2. Define the λ-*tube*

$$\mathcal{Y}_\lambda := \big\{ U \in W^{1,1}(0,1) : U(0) = U^\delta(0)\,,\ U(1) = U^\delta(1)\,, \\ \text{and } |U(t) - U^\delta(t)| \leq \lambda \text{ for } t \in (0,1) \big\}\,.$$

Step 3. We calculate the function $U_\lambda \in \mathcal{Y}_\lambda$ which minimizes the graph length, that is,

$$U_\lambda = \operatorname{argmin}_{U \in \mathcal{Y}_\lambda} \int_0^1 \sqrt{1 + (U')^2}\,.$$

Step 4. $u_\lambda := U'_\lambda$ is the outcome of the taut string algorithm.

In this approach, the amount of regularization depends on the parameter $\lambda > 0$. However, due to the properties of one-dimensional total variation regularization, the results are quite stable with respect to λ. We recall that one-dimensional total variation regularization satisfies a semi-group property: Repeated regularization first with a parameter $\lambda_1 > 0$ and then with a parameter $\lambda_2 > 0$ is the same as a single regularization step with a parameter $\lambda_1 + \lambda_2$ (see e.g. [20, Thm. 4.38]). In particular, this implies that for $\mu > \lambda$, the jump set of $\hat{S}_\mu$ is contained in the jump set of $\hat{S}_\lambda$. For this reason, the precise choice of the regularization parameter has, for modest values, no effect on the location of the most prominent jumps of $\hat{S}_\lambda$. In our implementations, we therefore chose the regularization parameter experimentally, using a sufficiently large parameter in order to remove most of the noise, but still retaining all the significant jumps.

Another possibility is to choose the smallest parameter λ for which the function $\hat{S}_\lambda$ is monotonically decreasing, since it is sufficient to find points where the scale of edges suddenly decreases.

Finally, we present some experimental results with Cameraman and Peppers image data, respectively in Figure 3 and Figure 4. To obtain these results, we first applied the *taut string algorithm* to filter the plots of the oscillating function S defined in (3.1). The plateaus between two discontinuities in the filtered function $\hat{S}_\lambda$ mark areas of a specific scale. Actually, there are more jumps detected than which we are separately displaying, as well shown in the filtered graphs with the figures, for the purpose of emphasising the scales detected by Algorithm 1, we just specify the most obvious jumps and figure them out part by part.

See Figure 3 and Figure 4, respectively. In Figure 3, the first discontinuity of S appears at iteration 17. There the most contrasty edges of the image —

the boundary of the hair and black coat of the photographer — are almost developed. The next significant jump appears at iteration 23, where the outline of the photographer has almost been formed. Then, at iteration 43, the structure of the cameraman has been caught. Finally, at iteration 64, also smaller details like the the camera are segmented.

In Figure 4 the situation is slightly different. The data contains many peppers where the contrast of the edges is rather similar. Thus the algorithm does not select complete pepper components but only parts of them on different peppers and associates them to a unique scale.

As a final example we apply the algorithm to the Cameraman, superimposed with Gaussian noise (see Figure 5). The results show the robustness of the method. Because of the noise, we used a larger regularization parameter in the taut string algorithm in order to select the different scales. Note that it was not necessary to change the regularization parameters α and β.

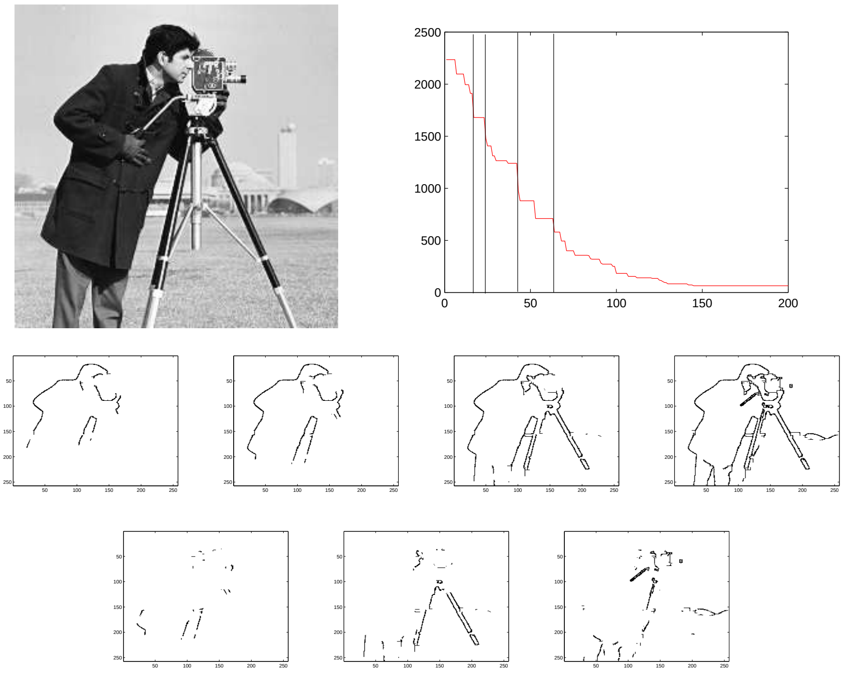

Fig. 3: Edge and scale detection in the Cameraman image. *Upper row, left:* The Cameraman image. *Upper row, right:* The TV filtered graph of function S. *Middle row:* The edge set K_k at the iterations k where the most prominent jumps occur (from left to right: iterations 17, 23, 43, and 64). *Lower row:* The difference between two subsequent edge sets.

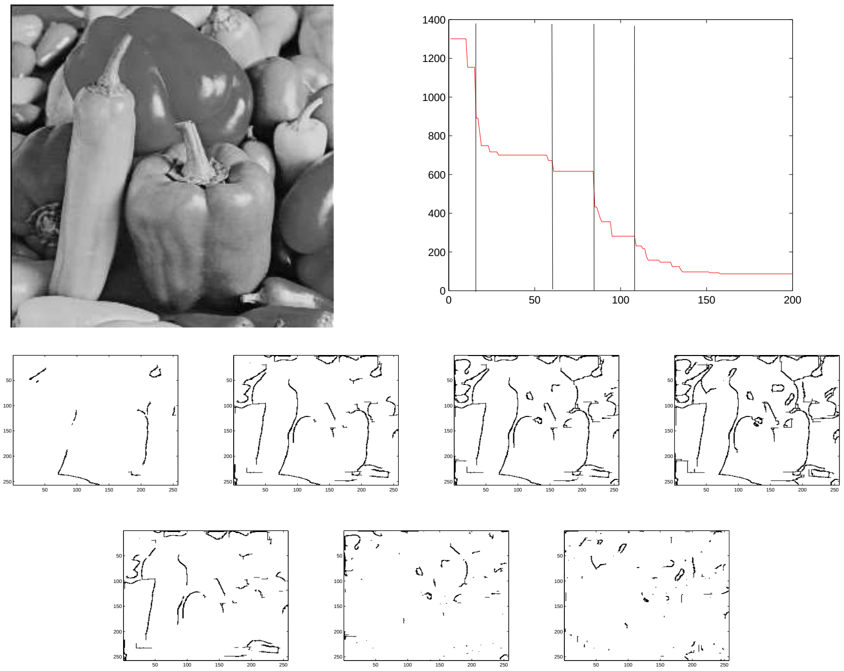

Fig. 4: Edge and scale detection of Peppers image. *Upper row, left:* The Peppers image. *Upper row, right:* The TV filtered graph of function S. *Lower row:* The edge set K_k at the iterations k where the most prominent jumps occur (from left to right: iterations 15, 60, 85, and 108). *Lower row:* The difference between two subsequent edge sets.

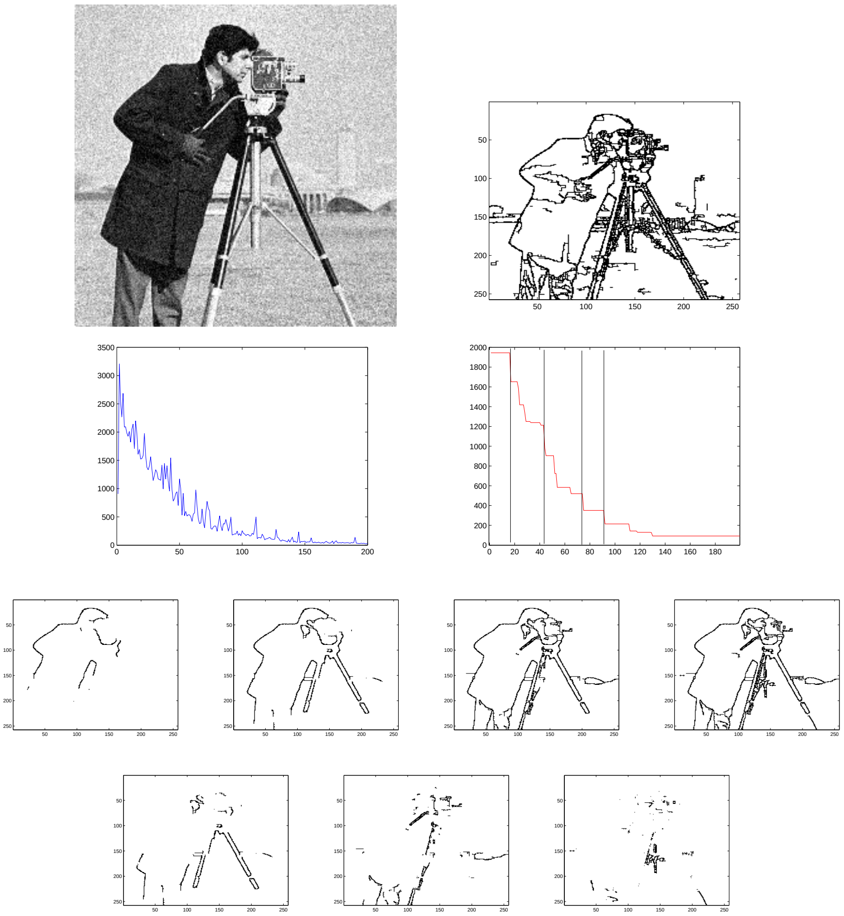

Fig. 5: Edge and scale detection in the presence of noise. *Upper row, left:* The noisy Cameraman image. *Upper row, right:* The edge set after 200 iterations. *Middle upper row, left:* The graph of the function S. *Middle upper row, right:* TV filtered graph of function S. *Middle lower row:* The edge set K_k at the iterations k where the most prominent jumps occur (from left to right: iterations 16, 43, 74, and 91). *Lower row:* The difference between two subsequent edge sets.

5 Conclusion

We presented a method to detect different scales of jumps across the boundary, using a recently developed algorithm [12] for approximating the Mumford–Shah algorithm using topological derivatives. This process is possible due to the locality of the algorithm, contrasting the globality of approaches like the Ambrosio–Tortorelli approximation. By considering the function S which represents the scale change of the edges, one can distinguish different parts of boundaries with different levels of jumps. For realistic images (also noisy), we applied total variation minimization using the taut string algorithm for a more stable scale separation. For future works, different norms can be considered for the function S, and there are possible improvements by adapting particular image features to the function $\hat{S}_\lambda$, or either considering to replace the current discrete count k which is the variable of the function S by a continuous parameter, as well, the impact of the roughness of edges on the scales separation is worth to be discussion.

Acknowledgements

The work of GD and OS has been supported by the Austrian Science Fund (FWF) within the national research networks Photoacoustic Imaging in Biology and Medicine, project S10505 and Geometry and Simulation, project S11704. All the authors want to express their thanks to the anonymous reviewers for their comments to improve the paper.

References

1. L. Ambrosio and V. M. Tortorelli. Approximation of functionals depending on jumps by elliptic functionals via Γ-convergence. *Comm. Pure Appl. Math.*, 43(8):999–1036, 1990.
2. D. Auroux, L. J. Belaid, and M. Masmoudi. A topological asymptotic analysis for the regularized grey-level image classification problem. *Math. Model. Numer. Anal.*, 41(3), 2007.
3. D. Auroux and M. Masmoudi. Image processing by topological asymptotic expansion. *J. Math. Imaging Vision*, 33(2), 2009.
4. A. Chambolle. Image segmentation by variational methods: Mumford and Shah functional and the discrete approximations. *SIAM J. Appl. Math.*, 55(3):827–863, 1995.
5. A. Chambolle. An algorithm for total variation minimization and applications. *J. Math. Imaging Vision*, 20(1–2):89–97, 2004.
6. A. Chambolle and P.-L. Lions. Image recovery via total variation minimization and related problems. *Numer. Math.*, 76(2):167–188, 1997.
7. T. Chan and L. Vese. Active Contours without Edges. In *IEEE Trans. Image Processing*, 10(2):266–277, 2001.
8. P. L. Davies and A. Kovac. Local extremes, runs, strings and multiresolution. *Ann. Statist.*, 29(1):1–65, 2001.
9. R. A. Feijóo, A. Novotny, C. Padra, and E. Taroco. The topological derivative for the Poisson problem. *Math. Mod. Meth. Appl. Sci.*, 13(12):1825–1844, 2003.

10. S. Garreau, P. Guillaume, and M. Masmoudi. The topological asymptotic for PDE systems: The elasticity case. *SIAM J. Control Optimiz*, 39(6):1756–1778, 2000.
11. M. Grasmair. The equivalence of the taut string algorithm and BV-regularization. *J. Math. Imaging Vision*, 27(1):59–66, 2007.
12. M. Grasmair, M. Muszkieta, and O. Scherzer. An approach to the minimization of the Mumford–Shah functional using Γ-convergence and topological asymptotic expansion. Preprint on ArXiv arXiv:1103.4722v1, University of Vienna, Austria, 2011.
13. Y. M. Jung and S. H. Kang and J. Shen. Multiphase Image Segmentation via Modica-Mortola Phase transition In *SIAM Applied Mathematics*, 67(5):1213–1232, 2007.
14. R. Kimmel, N. A. Sochen, and J. Weickert, editors. *Scale Space and PDE Methods in Computer Vision*, volume 3459 of *Lecture Notes in Computer Science*. Springer, 2005.
15. E. Mammen and S. van de Geer. Locally adaptive regression splines. *Ann. Statist.*, 25(1):387–413, 1997.
16. D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math.*, 42(5):577–685, 1989.
17. M. Muszkieta. Optimal edge detection by topological asymptotic analysis. *Math. Methods Appl. Sci.*, 19(11):2127–2143, 2009.
18. C. Pöschl and O. Scherzer. Characterization of minimizers of convex regularization functionals. In *Frames and operator theory in analysis and signal processing*, volume 451 of *Contemp. Math.*, pages 219–248. Amer. Math. Soc., Providence, RI, 2008.
19. L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1–4):259–268, 1992.
20. O. Scherzer, M. Grasmair, H. Grossauer, M. Haltmeier, and F. Lenzen. *Variational methods in imaging*, volume 167 of *Applied Mathematical Sciences*. Springer, New York, 2009.
21. J. Shen. A stochastic-variational model for Soft Mumford-Shah segmentation. In *International Journal of Biomedical Imaging*, ID92329, 2006.
22. J. Sokołowski and A. Żochowski. On topological derivative in shape optimization. *SIAM J. Control Optimiz*, 37(4):1251–1272, 1999.
23. G. Steidl, S. Didas, and J. Neumann. Relations between higher order TV regularization and support vector regression. In *[14]*, pages 515–527, 2005.