Numerical Invariantization for Morphological PDE Schemes

Martin Welk[1], Pilwon Kim[2], and Peter J. Olver[3]

[1] Mathematical Image Analysis Group
Faculty of Mathematics and Computer Science
Saarland University, 66041 Saarbrücken, Germany
welk@mia.uni-saarland.de
http://www.mia.uni-saarland.de
[2] Department of Mathematics
Ohio State University
Columbus, Ohio 43210, U.S.A.
pwkim@math.ohio-state.edu
[3] School of Mathematics
University of Minnesota
Minneapolis, MN 55455, U.S.A.
olver@math.umn.edu
http://www.math.umn.edu/~olver

Abstract. Based on a new, general formulation of the geometric method of moving frames, invariantization of numerical schemes has been established during the last years as a powerful tool to guarantee symmetries for numerical solutions while simultaneously reducing the numerical errors. In this paper, we make the first step to apply this framework to the differential equations of image processing. We focus on the Hamilton–Jacobi equation governing dilation and erosion processes which displays morphological symmetry, i.e. is invariant under strictly monotonically increasing transformations of gray-values. Results demonstrate that invariantization is able to handle the specific needs of differential equations applied in image processing, and thus encourage further research in this direction.

1 Introduction

Image filters based on partial differential equations play an important role in contemporary digital image processing. The field therefore has a need for efficient and accurate numerical algorithms for solving the PDEs that arise in applications.

The method of *invariantization* provides a general framework for designing numerical schemes for (ordinary and partial) differential equations [17, 12, 10] that preserve symmetries of the continuous-scale differential equation. The method is based on a new approach to the Cartan method of moving frames [4] that applies to completely general group actions, and has been extensively developed in the last few years [7, 18]. The invariantization process is based on a choice of cross-section to the symmetry group orbits, and careful selection of the cross-section can produce a more robust numerical scheme that is better able to handle rapid variations and singularities. So far, the invariantization technique has been studied for standard numerical schemes for ordinary

differential equations [10], as well as for a number of partial differential equations including the heat equation, the Korteweg–deVries equation, and Burgers' equation [11], with encouraging results.

In this paper, we will investigate the applicability of the invariantization framework in the context of image processing. This field of application poses special needs in that it requires in particular an accurate representation of sharp discontinuity-type structures such as edges. A variety of partial differential equations with discontinuity-preserving properties has been developed over the years but often numerical dissipation adversely affects their favorable theoretical properties. We are therefore especially interested in whether invariantization can contribute to reducing numerical dissipation effects and thereby improve the treatment of edges in images.

For our investigation, we select the Hamilton–Jacobi equations governing the morphological processes of dilation and erosion [1, 6]. They offer the advantage of combining formal simplicity with high relevance for image analysis – mathematical morphology being one of the oldest and most successful techniques in the field [15, 21] – and a particularly attractive symmetry property, namely the so-called morphological invariance. The latter is also a characteristic of many other image processing PDEs such as mean curvature motion [8, 9, 2], and the affine invariant morphological scale space [1, 20]. Thus, our present results can be viewed as a proof of concept for a wider application of the invariantization idea in this field.

2   Morphological PDEs

Dilation and erosion are the basic operations of mathematical morphology. Let S be a closed connected convex set containing zero. Dilation of a gray-value image u with S as structuring element then comes down to taking at each location the maximum of gray-values within the translated structuring element while erosion uses the minimum instead:

$$
\begin{aligned}
\text{dilation:} \quad & (u \oplus S)(x) := \max_{y \in S} u(x+y) \ , \\
\text{erosion:} \quad & (u \ominus S)(x) := \min_{y \in S} u(x+y) \ .
\end{aligned}
\tag{1}
$$

Dilation and erosion with disk-shaped structuring elements are closely related to the Hamilton–Jacobi partial differential equation

$$
u_t = \pm |\nabla u| \tag{2}
$$

where ∇u denotes the spatial gradient of u, i.e. $\nabla u = u_x$ in the 1D case, or $\nabla u = (u_x, u_y)^{\mathrm{T}}$ in the 2D case: Given the initial image u_0 at time $t = 0$, we evolve via (2) up to time t. In the case of the positive sign in (2) the resulting image u will be the dilation of u_0 with the disk $S = D_t = \{x \mid |x| \leq t\}$ as structuring element while in the case of the negative sign an erosion with the same structuring element results.

2.1   The Upwind Scheme

In spite of the simplicity of the PDE (2), its numerical evaluation remains a challenge. In image processing, one is particularly interested in the correct treatment of steep gradi-

ents which represent image edges. Under the Hamilton–Jacobi flow, these should propagate in space at constant speed without being blurred. Moreover, the partial differential equation (2) obeys a maximum–minimum principle which is also essential in image processing applications.

The simplest approach, a forward Euler discretization, with central spatial differences, generates oscillations in the vicinity of edges that violate the maximum–minimum principle; this is another manifestation of the general Gibbs phenomena observed in numerical approximations to discontinuous solutions, [14]. They can only be reduced but not eliminated by choosing very small time step sizes. Moreover, edges are smeared out as the number of iterations increases, and so the problem becomes even worse with smaller time steps.

An alternative scheme that avoids the oscillatory behavior and obeys the maximum–minimum principle is the upwind scheme [22]. Its idea is to discretize the first-order derivatives on the right-hand side of (2) by one-sided difference and switch between their possible directions depending on the local gradient direction, and hence on the information flow direction. In the case of 1D dilation, $u_t = |u_x|$, one step of the resulting explicit scheme with spatial grid size h and time step size τ then reads

$$u_i^{k+1} = u_i^k + \frac{\tau}{h} \max\{u_{i+1}^k - u_i^k, u_{i-1}^k - u_i^k, 0\} . \tag{3}$$

For time step sizes $\tau \leq h$ this scheme respects the maximum–minimum principle.

There are several ways to adapt this idea to the two-dimensional case. We defer these considerations until Subsection 3.3.

3   Morphological invariantization

In general, given a freely acting r-parameter transformation group acting on an m-dimensional space, one defines a moving frame by the choice of a cross-section to the group orbits, [7, 18]. In practice, one begins by writing out the group transformations as explicit functions of the coordinates $z = (z_1, \ldots, z_m)$ and the group parameters $\lambda = (\lambda_1, \ldots, \lambda_r)$. One then normalizes r of these expressions by equating them to well-chosen constants – typically either 0 or 1 – and solving for the group parameters in terms of the coordinates: $\lambda = \rho(z)$, which defines the moving frame map. The invariantization of any function, numerical scheme, etc. is then found by first writing out its transformed version and then replacing the group parameters by their moving frame formula. In particular, the invariantization of the coordinates z_i yields the fundamental invariants $I_i(z)$, with those corresponding to the r normalization coordinates being constant. The invariantization of any other function $F(z_1, \ldots, z_m)$ is then found by replacing each z_i by its corresponding invariant (constant or not), leading to the invariantized function $I(z) = F(I_1(z), \ldots, I_m(z))$. In particular, invariantization does not change a function that is already invariant under the group. This so-called Replacement Rule makes it particularly easy to convert (both mathematically and in pre-existing software packages) numerical schemes into invariant numerical schemes. The resulting schemes are guaranteed to be consistent with the underlying differential equations, since invariantization preserves consistency of numerical schemes. In fact, one of the key benefits

3

of the invariantization method is that it enables one to modify and tune existing schemes without affecting their consistency. In numerical applications, one selects the normalization coordinates and constants so as to try to eliminate as many of the error terms as possible; see [12, 10, 18] for further details.

3.1 Symmetry Group

The Hamilton–Jacobi PDE (2) that governs the processes of dilation and erosion displays one outstanding symmetry: It is invariant under *any* (differentiable) strictly monotonically increasing gray-value transformation [3]. This specific symmetry which is shared by a class of other PDEs relevant for image processing like mean curvature motion and affine curvature flow is called *morphological invariance*. PDEs with this symmetry can be re-formulated into intrinsic level set evolutions, i.e. curve or hypersurface evolutions of the level sets which depend on nothing else but the geometry of the evolving level sets themselves [19, 3]. Infinitesimal generators for this symmetry are given by $f(u)\partial_u$ for arbitrary differentiable functions $f(u)$.

From the viewpoint of the invariantization of numerical schemes, the morphological symmetry is special in that it involves the function values only, in contrast to the symmetries of many other differential equations that involve both the independent and the dependent variables. Moreover, it is a very rich symmetry since the group of strictly increasing differentiable maps of $\mathbb{R}$ is an infinite-dimensional Lie pseudogroup. Though an extension of the invariantization framework for the Lie pseudogroup case has been recently developed, [18], to simplify the constructions, we will restrict our attention to a particular one-dimensional subgroup.

To this end, we use the strictly monotonically increasing transformations

$$\tau_\lambda : [0, 1] \longrightarrow [0, 1], \qquad u \longmapsto \frac{\lambda u}{1 + (\lambda - 1)u} \tag{4}$$

where $\lambda \in \mathbb{R}^+$ is the group parameter. This family of functions on $[0, 1]$ forms a one-parameter Lie group with infinitesimal generator $u(1-u)\partial_u$, satisfying the group laws $\tau_\mu \circ \tau_\lambda = \tau_{\lambda\mu}$, $(\tau_\lambda)^{-1} = \tau_{1/\lambda}$.

3.2 The One-Dimensional Case

We want now to use the invariantization idea in order to improve the accuracy of numerical schemes for the 1D Hamilton–Jacobi equation $u_t = |u_x|$. With respect to image processing applications we are particularly interested in reducing numerical dissipation at edges.

The one-parameter Lie group selected in the previous subsection allows us to impose one equality constraint on the local numerical data. A closer look reveals that both the forward Euler scheme with central spatial differences and the upwind scheme are exact if the function u is linear in x. We want therefore to annihilate locally the second derivative u_{xx}.

While this idea is easy to carry out for the central difference scheme, it turns out that the numerical dissipation is in no way reduced. Thus, we turn our attention to

the upwind scheme. Since this scheme uses one-sided difference approximations for the first derivatives, the question arises which approximation of the second derivative should be used in the constraint that is to be enforced by invariantization. Since the first order derivative approximations can be considered as central differences located at inter-pixel positions $i \pm 1/2$, thus providing higher accuracy at these locations, we decide to use a four-pixel stencil centred at the same location for the second derivative.

Let us consider without loss of generality the case $u_x > 0$ in which the upwind scheme uses the right-sided derivative approximation. As approximation of the second derivative we then use $(u_{xx})_i \approx u_{i+2} - u_{i+1} - u_i + u_{i-1}$. For the invariantization at pixel i in time step k, we linearly transform the pixel values u_j^k, $j = i-1, i, i+1, i+2$, to $[0, 1]$ which gives $\tilde{u}_j^k$, and apply (4) to obtain $v_j^k = \tau_\lambda(\tilde{u}_j^k)$. Herein, the parameter $\lambda = \lambda_i^k > 0$ is to be determined, using the invariantization condition

$$v_{i+2}^k - v_{i+1}^k - v_i^k + v_{i-1}^k = 0 \ . \tag{5}$$

Inserting (4) into (5) gives

$$
\begin{aligned}
0 = \lambda \Big(& (\lambda + 1)^2 (-\tilde{u}_{i+2}^k \tilde{u}_{i+1}^k \tilde{u}_i^k + \tilde{u}_{i+2}^k \tilde{u}_{i+1}^k \tilde{u}_{i-1}^k + \tilde{u}_{i+2}^k \tilde{u}_i^k \tilde{u}_{i-1}^k - \tilde{u}_{i+1}^k \tilde{u}_i^k \tilde{u}_{i-1}^k) \\
& + 2(\lambda + 1)(\tilde{u}_{i+2}^k \tilde{u}_{i-1}^k - \tilde{u}_{i+2}^k \tilde{u}_i^k) + (\tilde{u}_{i+2}^k - \tilde{u}_{i+1}^k - \tilde{u}_i^k + \tilde{u}_{i-1}^k) \Big) \ .
\end{aligned}
\tag{6}
$$

This equation has exactly one positive solution if the sequence $u_{i-1}^k, u_i^k, u_{i+1}^k, u_{i+2}^k$ is strictly monotonic. If this is not the case, our one-parameter transformation group in fact does not contain a transformation that satisfies (5). Instead, $\lambda = 0$ is then calculated as largest solution of (6). We select therefore a small $\varepsilon > 0$ and use $\tilde{\lambda} = \max\{\lambda, \varepsilon\}$ in the algorithm. Whenever $\lambda < \varepsilon$, our invariantization is therefore imperfect, and the second derivative error term not completely annihilated. Still, the numerical error is reduced in these cases.

One time step for pixel i of a 1D signal reads therefore as follows.

1. Compute the one-sided derivative approximations $\Delta_{i,+}^k := u_{i+1}^k - u_i^k$, $\Delta_{i,-}^k := u_i^k - u_{i-1}^k$. If $\max\{\Delta_{i,+}^k, -\Delta_{i,-}^k, 0\} = 0$, let $u_i^{k+1} = u_i^k$ and finish. Otherwise, if $\Delta_{i,+}^k \geq -\Delta_{i,-}^k$, let $\sigma = +1$, else $\sigma = -1$. Let $\hat{u}_j := u_{i+j\sigma}^k$ for $j = -1, 0, 1, 2$.

2. Let
$$
\begin{aligned}
m &:= \min\{\hat{u}_j \mid j \in \{-1, 0, 1, 2\}\} \ , \\
M &:= \max\{\hat{u}_j \mid j \in \{-1, 0, 1, 2\}\} \ , \\
\tilde{u}_j &:= \frac{\hat{u}_j - m}{M - m} \ , \quad j \in \{-1, 0, 1, 2\} \ .
\end{aligned}
\tag{7}
$$

3. Compute the coefficients
$$
\begin{aligned}
a &:= \tilde{u}_2 \tilde{u}_1 \tilde{u}_0 - \tilde{u}_2 \tilde{u}_1 \tilde{u}_{-1} - \tilde{u}_2 \tilde{u}_0 \tilde{u}_{-1} + \tilde{u}_1 \tilde{u}_0 \tilde{u}_{-1} \ , \\
b &:= \tilde{u}_2 \tilde{u}_{-1} - \tilde{u}_1 \tilde{u}_0 \ , \\
c &:= \tilde{u}_2 - \tilde{u}_1 - \tilde{u}_0 + \tilde{u}_{-1}
\end{aligned}
\tag{8}
$$

and the transformation parameter

$$\lambda := 1 + \frac{b + \sqrt{b^2 + 4ac}}{a} \ . \tag{9}$$

Bound the transformation parameter via $\tilde{\lambda} := \max\{\lambda, \varepsilon\}$.

4. Transform the pixel values by

$$v_j := \tau_{\tilde{\lambda}}(\tilde{u}_j) , \quad j \in \{-1, 0, 1, 2\} . \tag{10}$$

5. Perform one step of the upwind scheme on the transformed data:

$$\tilde{v}_0 := v_0 + \frac{\tau}{h}(v_1 - v_0) . \tag{11}$$

6. Transform back:

$$u_i^{k+1} := m + (M - m)\tau_{1/\tilde{\lambda}}(\tilde{v}_0) . \tag{12}$$

It is easy to see that as for the unmodified upwind scheme, the maximum–minimum principle is guaranteed for the modified algorithm if the time step size fulfills $\tau < 1$.

3.3 The Two-Dimensional Case

In the two-dimensional situation there is a continuum of possible "upwind" directions. This adds complication to the discretization of first and second derivatives. While in the original upwind scheme an approximation of the gradient magnitude based on one-sided difference approximations of u_x and u_y works reasonably, experiments show that the invariantization via second derivatives is highly sensitive to misestimations of the second derivatives in gradient direction.
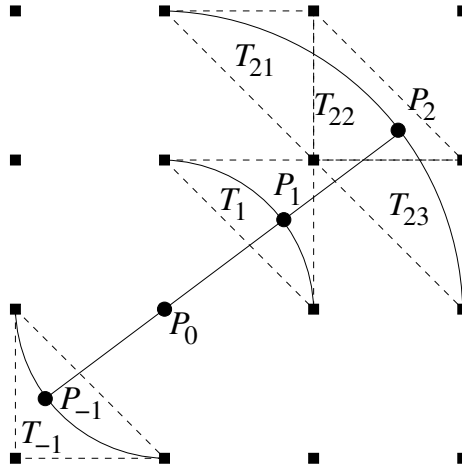

Fig. 1. Interpolation of a local 1-D subsample in gradient direction consisting of function values at P_{-1}, P_0, P_1, and P_2. The points P_{-1}, P_1, P_2 are located on circular arcs around P_0. P_{-1} is linearly interpolated within the triangle T_{-1}, P_1 within T_1, and P_2 within one of the triangles T_{21}, T_{22}, T_{23}.

However, since the 2D Hamilton–Jacobi flow at every single location is essentially a 1D process, we can directly build on our 1D algorithm in the following way. First,

6

we compute via u_x and u_y approximations in the spirit of classical 2D upwind scheme implementations the gradient direction. Then, we resample the needed pixels along this direction to obtain a 1D section that represents the problem at the given location. While in principle this could be done via bilinear interpolation within grid squares, we choose an interpolation within isosceles right triangles of side length 1 that experimentally represents the local features of the 1D section slightly better (see Fig. 1). To interpolate u for a point P on a 1D section through (i, j) in gradient direction, we use the triangle of grid points that encloses P and whose vertex has either maximal or minimal distance to (i, j) among the three corner points.

One time step for pixel (i, j) then reads as follows.

1. Compute

$$\Delta_{i,j;x+}^k := u_{i+1,j}^k - u_{i,j}^k \ , \qquad \Delta_{i,j;x-}^k := u_{i,j}^k - u_{i-1,j}^k \ ,$$
$$\Delta_{i,j;y+}^k := u_{i,j+1}^k - u_{i,j}^k \ , \qquad \Delta_{i,j;y-}^k := u_{i,j}^k - u_{i,j-1}^k \ . \tag{13}$$

If $\max\{\Delta_{i,j;x+}^k, -\Delta_{i,j;x-}^k, 0\} = 0$, let $s_x := 0$, $\Delta_x := 0$, else if $\Delta_{i,j;x+}^k \geq -\Delta_{i,j;x-}^k$, let $s_x := +1$, $\Delta_x := \Delta_{i,j;x+}^k$, else let $s_x := -1$, $\Delta_x := -\Delta_{i,j;x-}^k$. Proceed analogously to determine s_y and Δ_y.

2. If $\Delta_x = \Delta_y = 0$, let $u_{i,j}^{k+1} = u_{i,j}^k$ and finish. Otherwise, let

$$\sigma_x := \frac{s_x}{\sqrt{s_x^2 + s_y^2}} \ , \qquad \sigma_y := \frac{s_y}{\sqrt{s_x^2 + s_y^2}} \ . \tag{14}$$

3. Compute
$$\widehat{u}_l := u_{i+l\sigma_x, j+l\sigma_y}^k \ , \quad l = -1, 0, 1, 2 \ , \tag{15}$$
where inter-pixel values of u are linearly interpolated between three neighboring grid locations.

4. Apply steps 2–6 of the 1D algorithm to the 1D signal $\widehat{u}$, and assign the resulting value to $u_{i,j}^{k+1}$.

Though the calculation on the resampled 1D subsample involves inter-pixel sample values which are not present in the previous time step of the image, the maximum–minimum principle is still obeyed because the linear interpolation itself satisfies the maximum–minimum principle.

4 Experiments

4.1 One-Dimensional Case

To illustrate the effect of invariantization on a 1D example, Figure 2 shows the dilation of a single peak by the upwind scheme and our invariantized modification together with the theoretical solution. The higher sharpness of the invariantized scheme is clearly visible. We note that comparing with the theoretical result the propagation of the edge is slightly accelerated, an undesired effect that even increases for smaller time step sizes. The reason is that our scheme in its present form does not compensate for the bias in the treatment of regions of opposite curvature which is introduced by the use of one-sided derivative approximations. Since experimentally the effect is much smaller in the 2D case, we do not discuss remedies here.
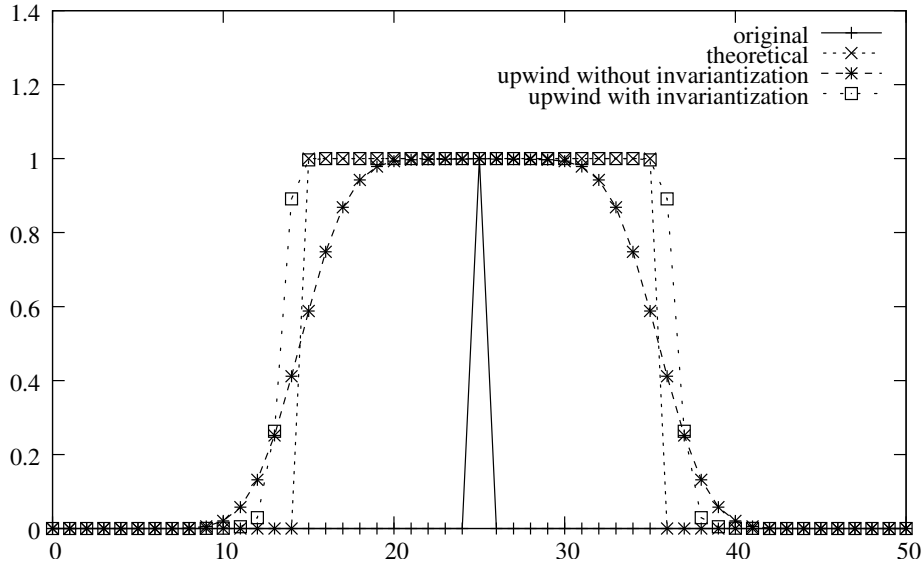
Fig. 2. 1D dilation of a single peak, 20 iterations with $\tau = 0.5$ of upwind scheme without and with invariantization. For comparison, the theoretical dilation result at evolution time $t = 10$ is also included.

4.2 Two-Dimensional Case

We demonstrate the 2D version of our algorithm with two experiments. First, Fig. 3 shows a test image featuring three discs, together with two stages of dilation evolution, for both the upwind scheme and our method. It is evident that the sharp boundaries of the expanding discs are preserved better by the invariantized scheme. The second stage of evolution demonstrates the correct handling of the merging between the objects.

At the same time, one can observe the reasonable degree of rotational invariance achieved by our method. This has been supported by choosing a smaller time step size than in the 1D case. Still, a close look suggests that a small amount of additional blur is added in diagonal directions due to the interpolation procedure used to obtain the 1D subsample.

A 1D section from the 2D evolution (slightly above the horizontal diameter of one circle, as indicated in Fig. 3) is shown in Fig. 4. The increased sharpness of the invariantized scheme is again visible; the interface between the bright and dark region attains a width of approx. four to five pixels, which is in accordance with the effective region of influence of each time step. This degree of edge blur remains essentially unchanged even after many more time steps.

The position of the expanded contour under an exact dilation with equal evolution time is also shown. Here, the speed of expansion of the bright regions is in good agreement with the theoretically derived speed, even with the smaller time step size. Besides this, the maximum–minimum stability is confirmed by Fig. 4.
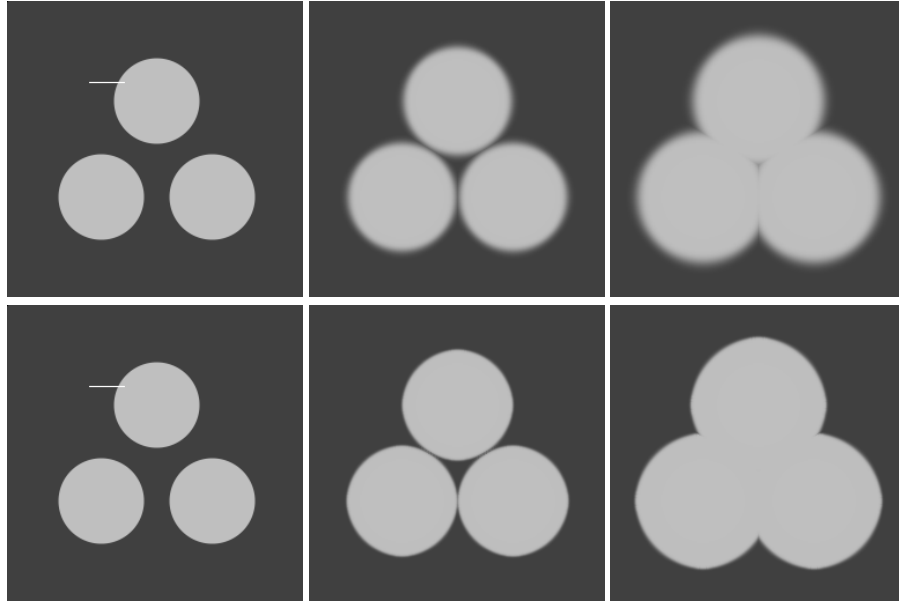
8

Fig. 3. Top, left: Original image (256×256 pixels) showing three discs. White line marks a 1D section shown in Fig. 4. **Top, middle:** Dilation by upwind scheme without invariantization, 100 iterations, time step $\tau = 0.1$. **Top, right:** Same but with 200 iterations. **Bottom row:** Same as above but with invariantized upwind scheme
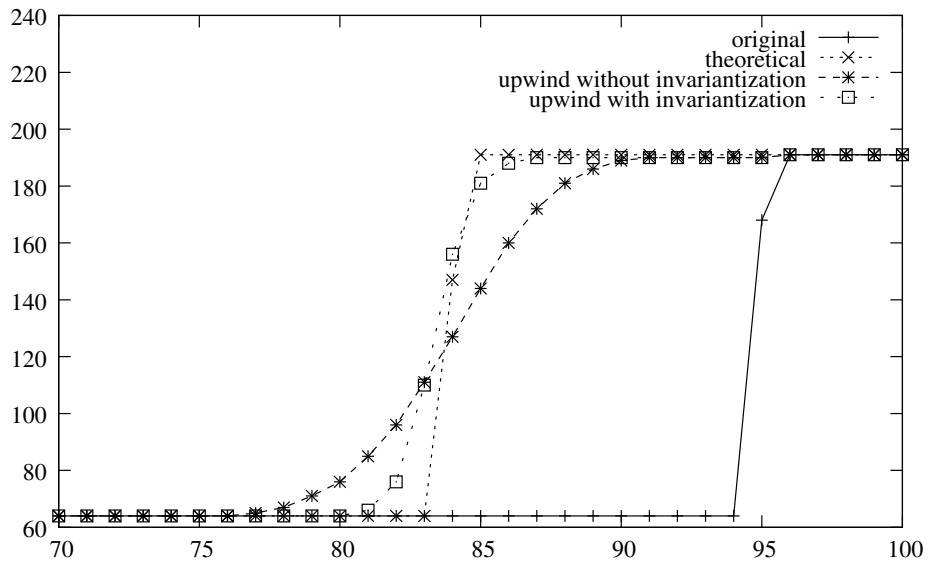

Fig. 4. Profiles of 2D dilation results along the line marked in Fig. 3. Original image, theoretical result of dilation at time $t = 10$, upwind scheme without and with invariantization, 100 iterations with $\tau = 0.1$

Fig. 5. Left: Original image (256 × 256 pixels). **Middle:** Dilation with invariantized upwind scheme, 50 iterations with $\tau = 0.1$ **Right:** Same with 150 iterations
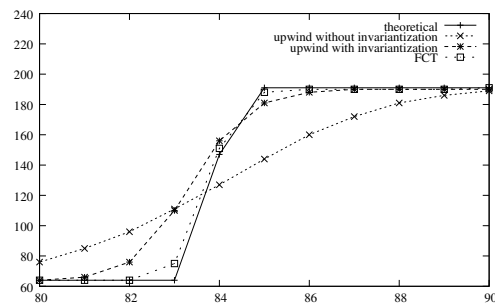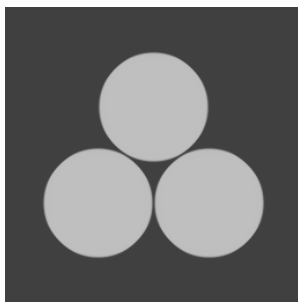

Fig. 6. Left: Dilation of Fig. 3 by flux-corrected transport (FCT) scheme, 20 iterations with $\tau = 0.5$ (provided by M. Breuß). **Right:** Central part of the dilated profiles from Fig. 4 and corresponding profile of the FCT result

Fig. 5 finally demonstrates the dilation process of a natural halftone image by our algorithm.

A comparison with another state-of-the-art numerical method for evaluating the Hamilton–Jacobi equation of dilation is shown in Fig. 6. The flux-corrected transport (FCT) scheme by Breuß and Weickert [5] relies on a direct modelling of, and compensation for, the numerical viscosity of the upwind scheme. Thereby, it achieves a higher degree of sharpness, with an interface width of only one to two pixels. Note that for the FCT scheme a larger time step size has been used. Since the two approaches exploit different aspects of the process, it will be worth conducting future research to look for ways how their respective advantages can be combined.

Erosion is equivalent to dilation of an inverted image and can therefore be performed in a completely analogous fashion by our method. Due to space limitations, we have not included an erosion example here.

5 Conclusion

We have demonstrated that the invariantization technique can be applied to the numerics of PDE-based image filters. It allows to raise the accuracy of numerical schemes

10

and also to reduce numerical problems that are particularly troublesome in image processing applications such as numerical blurring of edges. We have concentrated here on a particular interesting symmetry of PDEs occurring in image processing applications, namely morphological invariance. One direction of ongoing research is the transfer of these techniques to other image filtering schemes based on PDEs with invariance properties. Though our method already displays a reasonable rotational invariance, the high directional sensitivity of the process makes further improvements in this respect desirable. Also, combinations of the invariantization idea with conservation properties are of interest. Finally, by reducing the morphological symmetry to a one-parameter subgroup, it has not been fully used so far; a better exploitation of its potential is therefore also a topic of continued research.

Acknowledgement

References

1. L. Alvarez, F. Guichard, P.-L. Lions, and J.-M. Morel. Axioms and fundamental equations in image processing. *Archive for Rational Mechanics and Analysis*, 123:199–257, 1993.
2. L. Alvarez, P.-L. Lions, and J.-M. Morel. Image selective smoothing and edge detection by nonlinear diffusion. II. *SIAM Journal on Numerical Analysis*, 29:845–866, 1992.
3. F. Cao. *Geometric Curve Evolution and Image Processing*, volume 1805 of *Lecture Notes in Mathematics*. Springer, Berlin, 2003.
4. É. Cartan. *La méthode du repère mobile, la théorie des groupes continus, et les espaces généralisés.* Exposés de Géométrie no. 5, Hermann, Paris, 1935.
5. M. Breuß and J. Weickert. A shock-capturing algorithm for the differential equations of dilation and erosion. *Journal of Mathematical Imaging and Vision*, in press.
6. R.W. Brockett and P. Maragos. Evolution equations for continuous-scale morphological filtering. *IEEE Transactions on Signal Processing*, 42:3377–3386, 1994.
7. M. Fels and P.J. Olver. Moving coframes. II. Regularization and theoretical foundations. *Acta Appl. Math.*, 55:127–208, 1999.
8. M. Gage and R. S. Hamilton. The heat equation shrinking convex plane curves. *Journal of Differential Geometry*, 23:69–96, 1986.
9. M. Grayson. The heat equation shrinks embedded plane curves to round points. *Journal of Differential Geometry*, 26:285–314, 1987.
10. P. Kim. *Invariantization of Numerical Schemes for Differential Equations Using Moving Frames.* Ph.D. Thesis, University of Minnesota, Minneapolis, 2006.
11. P. Kim. *Invariantization of the Crank-Nicholson Method for Burgers' Equation.* Preprint, University of Minnesota, Minneapolis.
12. P. Kim and P.J. Olver. Geometric integration via multi-space. *Regular and Chaotic Dynamics*, 9(3):213–226, 2004.
13. R. Kimmel. *Numerical Geometry of Images.* Springer, Berlin 2004.
14. P.D. Lax. Gibbs Phenomena. *Journal of Scientific Computing*, 28:445–449, 2006.

15. G. Mathéron. *Random Sets and Integral Geometry.* Wiley, New York 1975.
16. P.J. Olver. *Applications of Lie Groups to Differential Equations.* Springer, New York 1986.
17. P.J. Olver. Geometric foundations of numerical algorithms and symmetry. *Applicable Algebra in Engineering, Communication and Computing*, 11:417–436, 2001.
18. P.J. Olver. A survey of moving frames. In H. Li, P.J. Olver, and G. Sommer, eds., *Computer Algebra and Geometric Algebra with Applications.* Volume 3519 of *Lecture Notes in Computer Science*, 105–138, Springer, Berlin, 2005.
19. G. Sapiro. *Geometric Partial Differential Equations and Image Analysis.* Cambridge University Press, Cambridge, UK, 2001.
20. G. Sapiro and A. Tannenbaum. Affine invariant scale-space. *International Journal of Computer Vision*, 11:25–44, 1993.
21. J. Serra. *Image Analysis and Mathematical Morphology.* Volume 1. Academic Press, London 1982.
22. J.A. Sethian. *Level Set Methods.* Cambridge University Press, Cambridge, UK, 1996.