

# Combining Curvature Motion and Edge-Preserving Denoising

Stephan Didas and Joachim Weickert

Mathematical Image Analysis Group  
Department of Mathematics and Computer Science  
Saarland University, Building E1 1  
66041 Saarbrücken  
{didas,weickert}@mia.uni-saarland.de

**Abstract.** In this paper we investigate a family of partial differential equations (PDEs) for image processing which can be regarded as isotropic nonlinear diffusion with an additional factor on the right-hand side. The one-dimensional analogues to this filter class have been motivated as scaling limits of one-dimensional adaptive averaging schemes. In 2-D, mean curvature motion is one of the most prominent examples of this family of PDEs. Other representatives of the filter class combine properties of curvature motion with the enhanced edge preservation of Perona-Malik diffusion. It becomes apparent that these PDEs require a careful discretisation. Numerical experiments display the differences between Perona-Malik diffusion, classical mean curvature motion and the proposed extensions. We consider, for example, enhanced edge sharpness, the question of morphological invariance, and the behaviour with respect to noise.

## 1 Introduction

Mean curvature motion and nonlinear diffusion filtering are classical methods in image processing for which feature directions in the image are important. Usually the two prominent directions for the local geometry in the image are the direction of the level line or isophote (along an edge) and its orthogonal direction, the flowline (across an edge). Choosing the amount of diffusion along these two directions appropriately gives a various range of different methods [1–3]. The earliest examples go back to the 1960s when Gabor proposed a method for deblurring of electron microscopy images [4, 5].

A prominent example where we only have diffusion along edges is mean curvature motion (MCM). Its theoretical properties have first been investigated by Gage, Hamilton and Huisken [6–8] in the 1980s. In the context of image processing, it first appeared in [9, 10]. Two nonlinear extensions of MCM are proposed by Alvarez et al. [11], or by Sapiro with the so-called self-snakes [12].

Nonlinear diffusion has first been proposed for image processing by Perona and Malik in 1990 [13]. Especially in its regularised variant by Catté et al. [14] it has become one of the standard tools for image denoising and simplification

in the meantime. The decomposition of the Perona-Malik approach in diffusion along gradient and normal direction is used to clarify its properties [15].

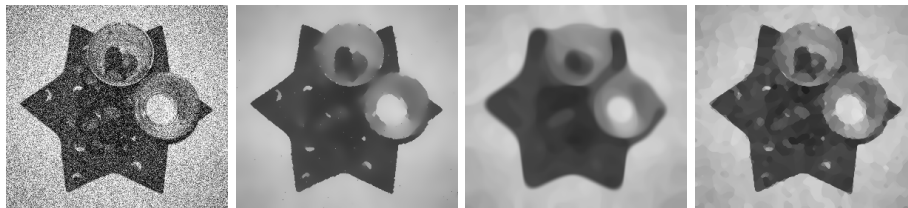
A general formulation for such filters which respect more general feature directions is given by Carmona and Zhong [16]. They discuss different ways to determine the local feature directions in practice, for example with second order derivatives or Gabor filters. The corresponding techniques are not only applied to grey value images, but there are also extensions to the vector-valued case by Tschumperlé and Deriche [17]. They use so-called oriented Laplacians, that means weighted sums of second derivatives of an image in two orthogonal directions.

The goal of the present paper is to investigate a class of PDEs for image processing which combine denoising capabilities of Perona-Malik filtering with curve shrinkage properties of mean curvature motion. The proposed model is the 2-D generalisation of a one-dimensional equation considered in [18] as scaling limit of adaptive averaging schemes. We are going to relate this filter class to previously known techniques and analyse its properties with numerical experiments.

The paper is organised as follows: The following Section 2 reviews some of the classical directional depending filters mentioned above to introduce the topic. Section 3 then introduces our model, which will be called *generalised mean curvature motion* (GMCM). In Section 4, we discuss some aspects of discretisation as this turns out to be a crucial question for this filter class. We display practical properties of all presented methods in Section 5 with several numerical examples and compare them to classical models. Section 6 concludes the paper with a summary and some questions of ongoing research.

## 2 Classical Directional Depending Filters

In this section, we review some well known filters which depend on the local feature directions. Fig. 1 gives an impression of the properties of several filtering methods. Nonlinear diffusion filtering as introduced by Perona and Malik [13]



**Fig. 1.** Classical PDE-based filters. *Left:* Original image,  $300 \times 275$  pixels, with Gaussian noise (standard deviation  $\sigma = 50$ ). *Second left:* Perona-Malik filtering,  $\lambda = 6$ ,  $t = 75$ . *Second right:* Mean curvature motion,  $t = 50$ . *Right:* Self-snakes,  $\lambda = 10$ ,  $t = 50$ .

uses the evolution of an image  $u$  under the PDE

$$\begin{aligned} u(\cdot, 0) &= f, \\ \partial_t u &= \operatorname{div}(g(|\nabla u|)\nabla u) \end{aligned} \quad (1)$$

where the given initial image is denoted by  $f$  and the evolving one by  $u$ . Usually homogeneous Neumann boundary conditions  $\partial_n u = 0$  are considered, i. e. the derivative in the normal direction of the boundary is zero. For all PDE methods in this paper, we have the same initial and boundary conditions, and we therefore do not state them explicitly for each equation in the following. Following the ideas in [15, 16] we decompose the diffusion into two parts acting in orthogonal directions. We consider the two local orientations of an image:

$$\eta := \frac{\nabla u}{|\nabla u|} = \frac{1}{\sqrt{u_x^2 + u_y^2}} \begin{pmatrix} u_x \\ u_y \end{pmatrix} \quad (2)$$

is the direction of the gradient or steepest ascent, that means across an edge in the image. Orthogonal to the gradient we have the direction of the level set

$$\xi := \frac{\nabla u^\perp}{|\nabla u^\perp|} = \frac{1}{\sqrt{u_x^2 + u_y^2}} \begin{pmatrix} -u_y \\ u_x \end{pmatrix}, \quad (3)$$

which points locally along an edge. In the following considerations we want to decompose a diffusion process into diffusion along and across image edges. For this reason we need the second derivatives of the image in the directions  $\xi$  and  $\eta$ , namely

$$u_{\xi\xi} = \frac{u_{xx}u_y^2 - 2u_xu_yu_{xy} + u_{yy}u_x^2}{u_x^2 + u_y^2} \quad \text{and} \quad (4)$$

$$u_{\eta\eta} = \frac{u_{xx}u_x^2 + 2u_xu_yu_{xy} + u_{yy}u_y^2}{u_x^2 + u_y^2}. \quad (5)$$

With these equations we follow Alvarez et al. [15] and decompose the Perona-Malik equation into two diffusion components acting in direction  $\xi$  and  $\eta$ :

$$\partial_t u = g(|\nabla u|) u_{\xi\xi} + (g(|\nabla u|) + g'(|\nabla u|)|\nabla u|) u_{\eta\eta} \quad (6)$$

We see that on the one hand, the factor  $g(|\nabla u|)$  can reduce the velocity of the diffusion close to an edge (when the gradient is large). On the other hand, the first derivative of  $g$  in the second summand makes backward diffusion in direction  $\eta$  possible. This gives the filter the capability of edge enhancement.

Starting from (6), Carmona and Zhong [16] proposed a more general evolution equation

$$\partial_t u = c(au_{\eta\eta} + bu_{\xi\xi}) \quad (7)$$

where the function  $c$  controls the whole amount of smoothing, and  $a$  and  $b$  weight this smoothing between the two feature directions. Carmona and Zhong let the

functions  $a, b$ , and  $c$  as given by the Perona and Malik equation (6) and focus on different ways to choose the local feature directions  $\xi$  and  $\eta$ . For example, they use eigenvectors of the Hessian of  $u$  or Gabor filters.

In contrast to their approach, we are going to modify the function  $c$  and leave  $\eta$  and  $\xi$  as given in (2) and (3). We will not focus on different choices of  $\xi$  and  $\eta$  in this paper, although it is clear that this could also be combined with our modified functions  $c$ . We are going to see with numerical examples that even small changes in  $c$  can change the filtering behaviour significantly.

Although this was not mentioned by Carmona and Zhong, mean curvature motion (MCM) can be obtained by choosing special parameters in their general filter class. It only performs smoothing in the direction  $\xi$  of the isophotes in the image:

$$\partial_t u = u_{\xi\xi} = |\nabla u| \operatorname{div} \left( \frac{\nabla u}{|\nabla u|} \right) . \quad (8)$$

There are some very useful properties of this technique [6–10]: First it is contrast invariant. Furthermore, it makes non-convex shapes convex and obeys a shape inclusion principle. Convex shapes are shrunk to circular points and finally vanish. In filtering time  $t = \frac{1}{2}r^2$ , a circle of radius  $r$  and everything inside has vanished. Nevertheless, mean curvature motion has the disadvantage to blur the edges during the evolution.

In the context of segmentation, Sapiro [12] proposed the so-called self-snakes which can be understood as nonlinear extension of mean curvature motion. The corresponding equation is

$$\partial_t u = |\nabla u| \operatorname{div} \left( g(|\nabla u|) \frac{\nabla u}{|\nabla u|} \right) \quad (9)$$

and allows for sharper edges. This can be explained by a decomposition of the equation in a curvature motion term and a shock filtering term making edges sharper.

### 3 Generalised Mean Curvature Motion

After reviewing some classical filtering methods in the last section, we now introduce the approach we will focus on in this paper. First we spend some words on the derivation of the general model, coming from adaptive averaging schemes. Then we interpret the appearing PDE with respect to the classical models. The study of important special cases obtained by several choices for the diffusivity-type function  $g$  in the model concludes this section.

#### 3.1 Derivation of the Model

The starting point for our derivations in this section is the consideration of adaptive averaging schemes

$$u_i^0 = f_i ,$$

$$u_i^{k+1} = \frac{g\left(\left|\frac{u_{i+1}^k - u_i^k}{h}\right|\right) u_{i+1}^k + g\left(\left|\frac{u_{i-1}^k - u_i^k}{h}\right|\right) u_{i-1}^k}{g\left(\left|\frac{u_{i+1}^k - u_i^k}{h}\right|\right) + g\left(\left|\frac{u_{i-1}^k - u_i^k}{h}\right|\right)} \quad (10)$$

in [18]. It is explained in detail there that a scaling limit of this averaging scheme leads to the so-called accelerated nonlinear diffusion equation

$$\partial_t u = \frac{1}{g(|\partial_x u|)} \partial_x (g(|\partial_x u|) \partial_x u) \quad . \quad (11)$$

The only difference of this equation to a classical Perona-Malik model is the factor  $\frac{1}{g(|\partial_x u|)}$  on the right-hand side. This factor is understood as acceleration of the process in the following sense: If we are in an almost constant region, the derivative of  $u$  is small, and the factor is close to 1. This does not change the evolution very much. On the other hand, at the position of an edge, we have a large derivative of  $u$ , and the factor is becoming much larger than 1. This leads to a higher evolution velocity close to an edge.

There is no unique way to generalise (11) to two or more dimensions. As described in [18], considering a 2-D weighted averaging scheme as starting point and taking the scaling limit leads to an anisotropic diffusion equation including a diffusion tensor.

In this paper, we generalise (11) in a different way to 2-D: We replace the first derivative of  $u$  in (11) by a gradient and the outer derivative by the divergence. This directly leads to the PDE

$$\partial_t u = \frac{1}{g(|\nabla u|)} \operatorname{div} (g(|\nabla u|) \nabla u) \quad , \quad (12)$$

which will be called *generalised mean curvature motion (GMCM)* here. To justify this name, consider the special case  $g(s) = \frac{1}{s}$  to obtain the standard mean curvature motion equation (8). This already indicates that the additional factor on the right-hand side changes the behaviour compared to the Perona-Malik model more than in the 1-D case.

### 3.2 Interpretation

From the decomposition of the Perona-Malik filter in (6), we immediately derive that generalised mean curvature motion (12) can be decomposed as

$$\partial_t u = u_{\xi\xi} + \left(1 + \frac{g'(|\nabla u|)|\nabla u|}{g(|\nabla u|)}\right) u_{\eta\eta} \quad (13)$$

$$= u_{\xi\xi} + a(|\nabla u|) u_{\eta\eta} \quad . \quad (14)$$

That means we have a mean curvature motion equation with an additional diffusive component in orthogonal direction  $\eta$  which is steered by the factor  $a(s) := 1 + \frac{g'(s)s}{g(s)}$ . As argument  $s$ , the factor depends on the norm of the gradient  $|\nabla u|$ . We will discuss later how the choice of  $g$  influences the behaviour of

this factor  $a(s)$ . The basic idea is that the filter performs shrinkage of level lines in the sense of mean curvature motion while the second term keeps edges sharp during the evolution.

There is also another way of understanding the process: Having the equation  $\Delta u = u_{\xi\xi} + u_{\eta\eta}$  in mind, we can rewrite this as

$$\partial_t u = \Delta u + \frac{g'(|\nabla u|)|\nabla u|}{g(|\nabla u|)} u_{\eta\eta} . \quad (15)$$

In this form, we see that the generalised mean curvature motion can be understood as linear diffusion with an additional shock term for edge enhancement. While classical Perona-Malik filtering slows down the linear diffusion part near edges by the factor  $g(|\nabla u|)$ , the velocity of this part is constant for generalised mean curvature motion.

### 3.3 Choices for the Function $g$

After specifying the general framework, we now focus on several choices of the function  $g$  and give some first insight in the expected behaviour of the corresponding methods.

**Perona-Malik diffusivity.** Let us choose the classical diffusivity function  $g(s) = \left(1 + \frac{s^2}{\lambda^2}\right)^{-1}$  proposed by Perona and Malik [13]. This diffusivity is especially interesting because it is capable of switching between forward and backward diffusion adaptively. In this case we have

$$a(s) = 1 + \frac{g'(s)s}{g(s)} = 1 - 2\frac{s^2}{s^2 + \lambda^2} \quad (16)$$

which immediately shows that  $-1 \leq a(s) \leq 1$  for all  $s \in \mathbb{R}$ . In a region where  $|\nabla u|$  is small, we have forward diffusion. That means the whole process (13) acts like linear diffusion there. Close to an edge, we have forward diffusion along the edge and backward diffusion across the edge. This explains the edge-preserving behaviour which can be observed at the practical results in Section 5.

**An example with unbounded backward diffusion.** Another frequently used diffusivity function is  $g(s) = \exp\left(-\frac{s^2}{2\lambda^2}\right)$  which has also been proposed by Perona and Malik [13]. In the classical nonlinear diffusion approach, it has the same properties as the function discussed above. In our case we obtain  $a(s) = 1 - \frac{s^2}{\lambda^2}$ . We have  $a(s) \leq 1$  for all  $s \in \mathbb{R}$ , but  $a$  is not bounded from below. That means in theory there would be no limit for the amount of backward diffusion in a pixel where  $|\nabla u|$  is very large. We see that similar diffusion properties do not imply a similar behaviour in the corresponding GMCM model. Nevertheless, this special example is of rather theoretical interest, since for realistic values of  $|\nabla u|$  and  $\lambda$ , the values  $\exp(|\nabla u|^2/\lambda^2)$  and  $\exp(-|\nabla u|^2/\lambda^2)$  differ by so many orders of magnitude that a numerical treatment gets very complicated.

**Special case: Constant diffusion velocity in both directions.** So far, we have chosen the amount of diffusion in direction  $\eta$  adaptively depending on

the gradient magnitude of the evolving image  $|\nabla u|$ . Now we consider the case that the diffusion in direction  $\eta$  has a constant velocity. This is equivalent to

$$a(s) = 1 + \frac{g'(s)s}{g(s)} = c \in \mathbb{R} . \quad (17)$$

We see that this condition is satisfied for the family of functions  $g(s) = \frac{1}{s^p}$  for  $p > 0$  where we have  $a(s) = 1 - p$ . The corresponding equation is given by

$$\partial_t u = |\nabla u|^p \operatorname{div} \left( \frac{\nabla u}{|\nabla u|^p} \right) = u_{\xi\xi} + (1 - p) u_{\eta\eta} . \quad (18)$$

For  $p = 1$ , we have the special case of mean curvature motion. In the experimental section, we are going to take a closer look at the behaviour of this filtering family for several integer values of  $p$ .

**Historical remark.** A certain special case of this family of methods has been proposed already in 1965 by Gabor in the context of electron microscopy [4]. Later on, his approach has been reviewed and brought into the context of contemporary image analysis [5]. Rewriting his approach in our notation gives the equation

$$u = f - \frac{\mu^2}{2} \left( f_{\eta\eta} - \frac{1}{3} f_{\xi\xi} \right) \quad (19)$$

for an initial image  $f$  and a filtered version  $u$ . The quantity  $\mu$  is derived from the application. We rewrite this equation as

$$\frac{6}{\mu^2} (u - f) = f_{\xi\xi} - 3f_{\eta\eta} . \quad (20)$$

The left-hand side of (20) can be interpreted as finite difference approximation of a time-derivative. That means, Gabor's approach (19) can be seen as one step in an Euler forward time-discretisation of (18) with  $p = 4$  and time step size  $\tau = \frac{\mu^2}{6}$ . Due to limited computational tools, the approach was rather theoretically motivated than used in practice at that time.

## 4 Discretisation

This section describes one possibility of discretising generalised mean curvature motion (13). In our first practical examples, it turns out that the question of finding a suitable discretisation is very important for this kind of equations.

Let  $h_1, h_2 > 0$  be the pixel distance in  $x$ - and  $y$ -direction and  $\mathcal{N}_d(i)$  the indices of the direct neighbouring pixels in direction  $d$  to the pixel with index  $i$ . Let  $u_i^k$  denote the grey value of pixel  $i$  at the time step  $k$ . We start with the grey values of the given initial image  $u_i^0 := f_i$ . Let further  $g_i^k \approx g(|\nabla u(x_i)|)$  denote an approximation to the weighting function evaluated at pixel  $i$ . We have used the approximation

$$|\nabla u(x_i)| \approx \sqrt{\sum_{d=1}^2 \sum_{j \in \mathcal{N}_d(i)} \frac{(u_j - u_i)^2}{2h_d^2}} . \quad (21)$$

This approximation yields better results for (13) than standard central differences. Similar to [19] we use a finite difference scheme with harmonic averaging of the diffusivity approximations:

$$u_i^{k+1} = \begin{cases} u_i^k & \text{if } g_i^k = 0 \\ u_i^k + \frac{1}{g_i^k} \sum_{d=1}^2 \sum_{j \in \mathcal{N}_d(i)} \frac{2}{\frac{1}{g_j^k} + \frac{1}{g_i^k}} \frac{u_j^k - u_i^k}{h_d^2} & \text{else} \end{cases} \quad (22)$$

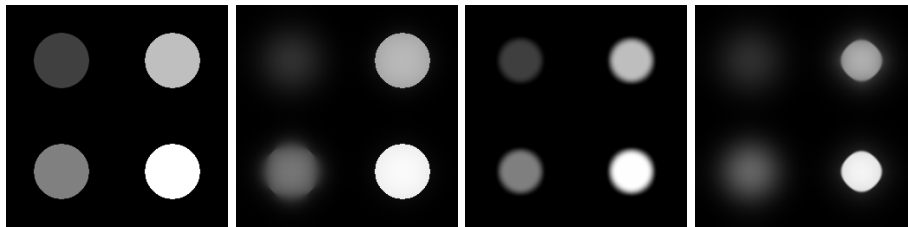
Why this scheme is very stable in practice can be seen by a simple equivalent reformulation of the scheme for  $g_i^k \neq 0$ :

$$u_i^{k+1} = u_i^k + \sum_{d=1}^2 \sum_{j \in \mathcal{N}_d(i)} \frac{2g_j^k}{g_j^k + g_i^k} \frac{u_j^k - u_i^k}{h_d^2}. \quad (23)$$

Under the assumption that  $g$  is a non-negative function, we have  $0 \leq \frac{g_j^k}{g_j^k + g_i^k} \leq 1$ . This allows us to see that for sufficiently small time step size  $\tau \leq \frac{1}{8}$  the iteration step (23) yields a convex combination of grey values from the old time step. We conclude that  $\min_j f_j \leq u_i^k \leq \max_j f_j$  for all  $k \in \mathbb{N}$  and all  $i$ , i. e. the process satisfies a maximum-minimum principle:

## 5 Numerical Experiments

In this section we study the properties of generalised mean curvature motion with some practical examples. In Fig. 2 we compare the results of Perona-Malik filtering with mean curvature motion and generalised mean curvature motion. It is clearly visible that GMCM offers a combination of the properties of MCM with Perona-Malik filtering: On the one hand, the contrast parameter  $\lambda$  gives us the opportunity to distinguish between smoothed and preserved sharp edges as known from Perona-Malik filtering. On the other hand, the objects are shrunken to points and vanish at finite time as known from mean curvature motion.



**Fig. 2.** Comparison of different filtering techniques. *Left:* Original image,  $256 \times 256$  pixels. *Second left:* Perona-Malik filtering,  $\lambda = 10$ . *Second right:* Mean curvature motion. *Right:* Generalised mean curvature motion (12) with  $g(s) = (1 + s^2/\lambda^2)^{-1}$ ,  $\lambda = 10$ . Stopping time in all examples:  $t = 200$ .



In our second experiment, we compare the behaviour of equation (18) for different values of  $p$ . Fig. 3 shows the results of the application to the same test image. We see that  $p = 1$  yields blurred results while  $p \geq 2$  leads to sharp edges. Some basic properties of mean curvature motion are also satisfied here: At this example we see that non-convex objects are getting convex and shrink in finite time. Further, for larger  $p$  it is possible that corners are also kept longer in the iterations, the process of making objects circular is slowed down. That means objects are getting smaller during evolution while the shape is preserved longer than for mean curvature motion.

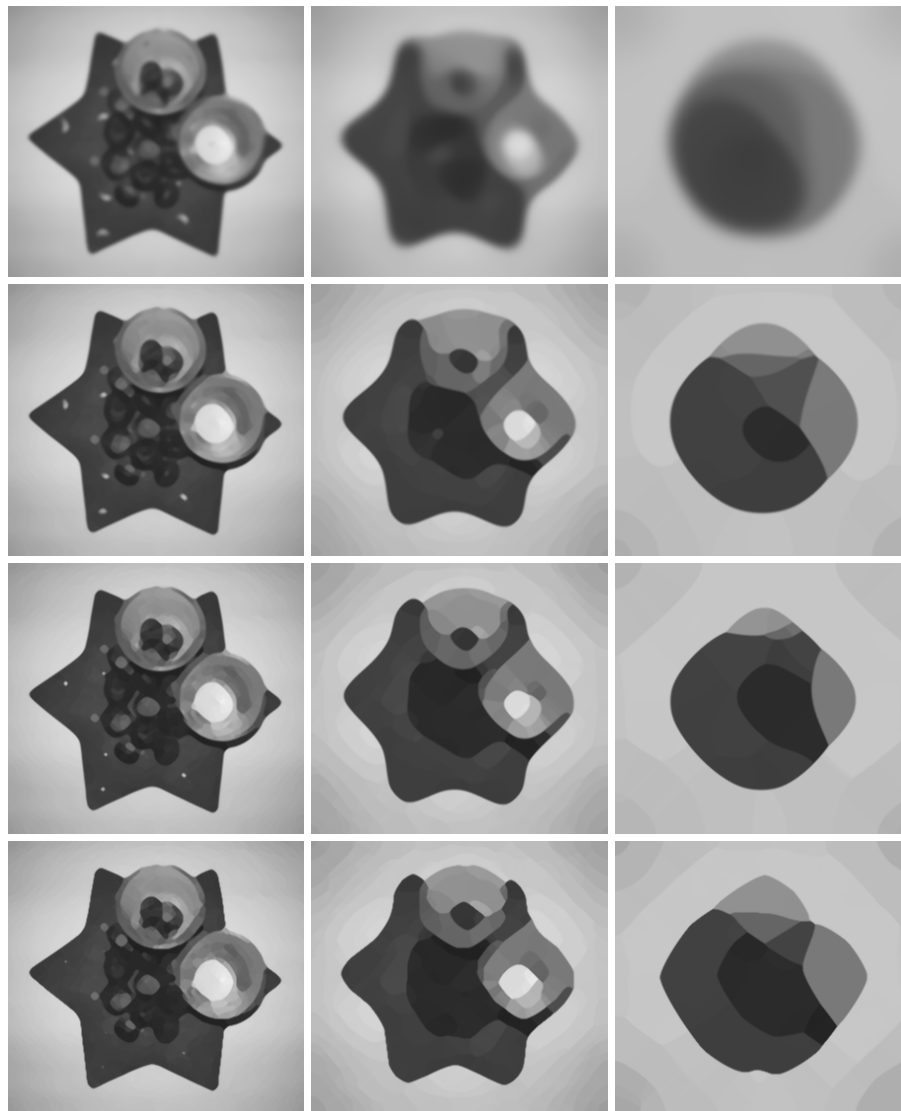
We have already mentioned that one important property of mean curvature motion is the morphological invariance. We use a test image composed out of four circles with different contrast to the background (see Fig. 2) to determine the contrast dependence of generalised mean curvature motion (18). We see that for  $p = 2, 4, 6, 10$  the four circles in one filtering result always have very similar size. This means, for constant regions, the contrast in this example does hardly influence the shrinkage time. We know from Fig. 3 that these processes tend to segment images into constant regions after a few steps. Further we notice that the stopping times for shrinkage of the circles changes strongly with  $p$ . Our experience which is also confirmed by a larger number of experiments is that the stopping time is smallest for  $p = 4$  and increases rapidly for larger  $p$ .

In Fig. 5, we see how joint denoising and curve shrinking is possible with generalised mean curvature motion. In this example, it is possible to obtain sharp edges even for a highly noisy test image. At the same time, the process shrinks circles with a comparable velocity to mean curvature motion. We see that self-snakes also denoise the image, but do not shrink it even for a larger stopping time.

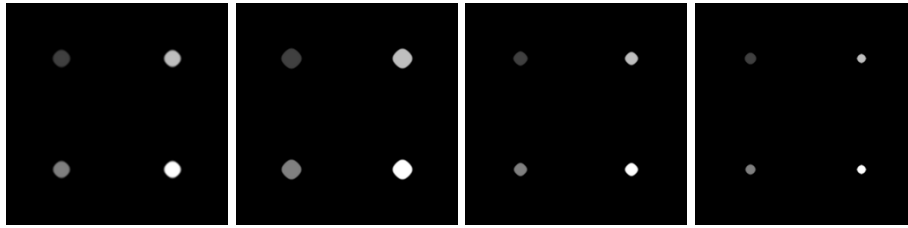
## 6 Conclusions

We have investigated a family of partial differential equations which is motivated by the consideration of adaptive averaging schemes. This family comprises mean curvature motion as prominent special case and earns properties related to level line shrinkage from this filter. On the other hand, its close relationship to Perona-Malik filtering explains that it is capable of smoothing edges selectively with a contrast parameter. Numerical examples have shown the properties of several representants of the filter family. It is clearly visible that linear generalised mean curvature motion yields much sharper results than classical mean curvature motion and keeps its interesting properties. Nonlinear generalised mean curvature motion combines Perona-Malik filtering with curve shrinkage. Questions of ongoing research include other ways to discretise the equations without the harmonic mean as well as theoretical properties such as shape inclusion and shrinkage times.

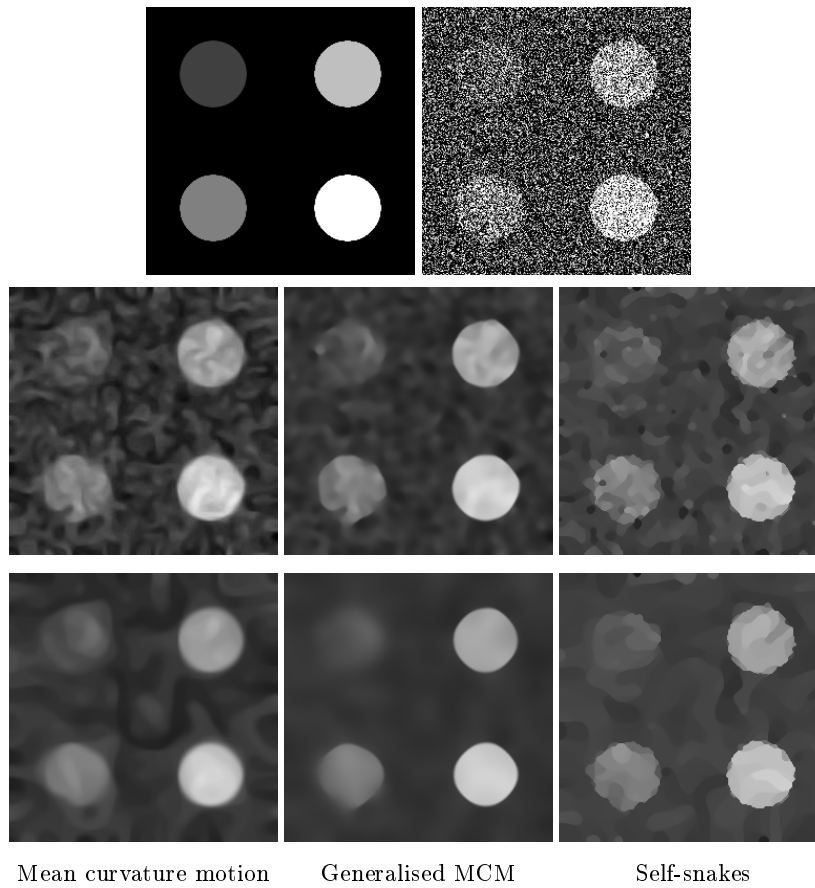
**Acknowledgements.** We gratefully acknowledge partly funding by the *Deutsche Forschungsgemeinschaft (DFG)*, project WE 2602/2-2.



**Fig. 3.** Comparison of the evolution (18) for different values of  $p$ . Rows from top to bottom:  $p = 1, 2, 4, 10$ . Left column:  $t = 5$ . Middle column:  $t = 100$ . Right column:  $t = 1000$ .



**Fig. 4.** Contrast dependency of a discrete version for the constant generalised mean curvature motion (18). *Left:*  $p = 2, t = 375$ . *Second left:*  $p = 4, t = 312.5$ . *Second right:*  $p = 6, t = 375$ . *Right:*  $p = 10, t = 625$ .



**Fig. 5.** Joint denoising and curvature motion. *Top left:* Original image,  $256 \times 256$  pixels. *Top right:* Original image with Gaussian noise, standard deviation  $\sigma = 200$ . *First column:* MCM,  $t = 12.5, 50$ . *Second column:* Generalised mean curvature motion (12),  $g(s) = (1 + s^2/\lambda^2)^{-1}$ ,  $t = 12.5, 50$ . *Third column:* Self-snakes,  $\lambda = 10, t = 50, 100$ .

## References

1. Sapiro, G.: Geometric Partial Differential Equations and Image Analysis. Cambridge University Press, Cambridge, UK (2001)
2. Kimmel, R.: Numerical Geometry of Images: Theory, Algorithms, and Applications. Springer, New York (2003)
3. Cao, F.: Geometric Curve Evolution and Image Processing. Volume 1805 of Lecture Notes in Mathematics. Springer, Berlin (2003)
4. Gabor, D.: Information theory in electron microscopy. *Laboratory Investigation* **14** (1965) 801–807
5. Lindenbaum, M., Fischer, M., Bruckstein, A.: On Gabor's contribution to image enhancement. *Pattern Recognition* **27** (1994) 1–8
6. Gage, M.: Curve shortening makes convex curves circular. *Inventiones Mathematicae* **76** (1984) 357–364
7. Gage, M., Hamilton, R.S.: The heat equation shrinking convex plane curves. *Journal of Differential Geometry* **23** (1986) 69–96
8. Huisken, G.: Flow by mean curvature of convex surfaces into spheres. *Journal of Differential Geometry* **20** (1984) 237–266
9. Osher, S., Sethian, J.A.: Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations. *Journal of Computational Physics* **79** (1988) 12–49
10. Kimia, B.B., Tannenbaum, A., Zucker, S.W.: Toward a computational theory of shape: an overview. In Faugeras, O., ed.: *Computer Vision – ECCV '90*. Volume 427 of *Lecture Notes in Computer Science*. Springer, Berlin (1990) 402–407
11. Alvarez, L., Lions, P.L., Morel, J.M.: Image selective smoothing and edge detection by nonlinear diffusion ii. *SIAM Journal on Numerical Analysis* **29** (1992) 845–866
12. Sapiro, G.: Vector (self) snakes: a geometric framework for color, texture and multiscale image segmentation. In: *Proc. 1996 IEEE International Conference on Image Processing*. Volume 1., Lausanne, Switzerland (1996) 817–820
13. Perona, P., Malik, J.: Scale space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **12** (1990) 629–639
14. Catté, F., Lions, P.L., Morel, J.M., Coll, T.: Image selective smoothing and edge detection by nonlinear diffusion. *SIAM Journal on Numerical Analysis* **29** (1992) 182–193
15. Alvarez, L., Guichard, F., Lions, P.L., Morel, J.M.: Axioms and fundamental equations of image processing. *Archive for Rational Mechanics and Analysis* **123** (1993) 199–257
16. Carmona, R.A., Zhong, S.: Adaptive smoothing respecting feature directions. *IEEE Transactions on Image Processing* **7** (1998) 353–358
17. Tschumperlé, D., Deriche, R.: Vector-valued image regularization with PDE's: A common framework for different applications. *IEEE Transactions on Image Processing* **27** (2005) 1–12
18. Didas, S., Weickert, J.: From adaptive averaging to accelerated nonlinear diffusion filtering. In Franke, K., ed.: *Pattern Recognition*. Volume 4174 of *Lecture Notes in Computer Science*, Springer, Berlin (2006) 101–110
19. Weickert, J.: Applications of nonlinear diffusion in image processing and computer vision. *Acta Mathematica Universitatis Comenianae* **LXX** (2001) 33–50