The Beltrami-Mumford-Shah Functional

Nir Sochen and Leah Bar

Department of Mathematics,
Tel-Aviv University
Tel-Aviv, 69978
Israel

Abstract. We present in this paper a unifying generalization of the Mumford-Shah functional, in the Ambrosio-Totorelli set up, and the Beltrami framework. The generalization of the Ambrosio-Tortorelli is in using a diffusion tensor as an indicator of the edge set instead of a function. The generalization of the Beltrami framework is in adding a penalty term on the metric such that it is defined dynamically from minimization of the functional.

We show that we are able, in this way, to have the benefits of true anisotropic diffusion together with a dynamically tuned metric/diffusion tensor. The functional is naturally defined in terms of the vielbein-the metric's square root. Preliminary results show improvement on both the Beltrami flow and the Mumford-Shah flow.

Keywords: Inhomogeneous diffusion, anisotropic diffusion, Mumford-Shah functional, Ambrosio-Tortorelli functional, Beltrami framework.

1 Introduction

The seminal work of Mumford and Shah[6] was a breakthrough, both conceptually and technically. From conceptual standpoint it reveled the need to unify the de-noising and edge detection problems into one problem where the two tasks are simultaneously solved. Technically it introduced a continuous functional to give the conceptual understanding a mathematical language and used calculus of variations to derive partial differential equations for the solution. It was interpreted later, via the relation to statistical inference ideas, as the prior on images that favors piecewise smooth functions over other possible functions.

In the early nineties the concept of inhomogeneous diffusion, coined "anisotropic diffusion" by Perona and Malik [7], gained popularity. With this method (which was discovered earlier in an independent manner in mathematical physics by Rosenau [8]) it was possible to construct a controlled non-linear filtering that reduces noise on one hand and conserves the sharpness of the image on the other. The relation between "anisotropic diffusion" and the variational approach became clearer after the "Total Variation" (TV) functional was introduced by Rudin, Osher and Fatemi [9] and was later generalized by Faugeras and Deriche in the Φ-formalism [5]. In these inhomogeneous diffusion methods

A.M. Bruckstein et al. (Eds.): SSVM 2011, LNCS 6667, pp. 183–193, 2012.

the form of the local diffusion coefficient is predefined in advance. Usually the diffusion coefficient is given as a known function of the amplitude of the local gradient. One can show that, similarly to the Mumford-Shah approach, all these methods treat images as functions and impose piecewise smoothness as a prior.

Inhomogeneous diffusion was linked to the Mumford-Shah functional by the seminal work of Ambrosio and Tortorelli [1]. In that approach the set of discontinuities in the image is represented by an auxiliary, soft indicator, function. This function serves as a local diffusion coefficient in the Euler-Lagrange or gradient descent equations for the image. The difference from other inhomogeneous diffusion methods is in the fact that this diffusion coefficient is determined *dynamically* by the minimization of the functional. This dynamic choice of discontinuities position and magnitude enhances the performance of de-noising/de-blurring algorithms [3,4].

Towards the end of the nineties another distinction, and consequently, an advancement was achieved. Weickert in the "Coherence-Enhancing diffusion" (CD) [11] and Sochen et al. in the "Beltrami flow" (BF) [10] introduced true anisotropic diffusion where the local diffusion function was replaced with a full rank diffusion tensor. The latter was linked in the Beltrami framework to a Riemannian metric. In this approach the image is not a function any more but a Riemannian manifold and the anisotropic diffusion is a consequence of diffusion of the image on an image-induced non-flat manifold. Both in the CD and in the BF the diffusion tensor's form is given in advance either as a variant of the structure tensor in CD or as the induced metric in the BF.

It is the aim of this paper to generalize the Beltrami framework and the Mumford-Shah approach by extending the respective functionals to a unifying one. The starting point is the seminal work of Ambrosio and Tortorelli [1]. We extend their approach that treat the image and it set of discontinuities as two different *dynamical* variables that should be optimized by the same functional. We present in this work a functional where the (color) image and the diffusion tensor are treated both as dynamical variables. We interpret the diffusion tensor as a metric and end up with an extension of the Beltrami framework.

The paper is organized as follows: We review inhomogeneous diffusion methods and its derivation as a minimization of a functional in Section 1. The Mumford-Shah functional and the Ambrosio-Tortorelli approach are presented in Section 2. In that section we will also point out to the relation of the minimization of the Ambrosio-Tortorelli's functional to inhomogeneous diffusion. Section 3 presents anisotropic diffusion via the Coherent diffusion and the Beltrami frameworks. Generalizing the anisotropic diffusion in an "Ambrosio-Tortorelli like" functional is presented In section 4. The generalization of the Mumford-Shah functional is find to generalize the Polyakov action of the Beltrami framework at the same time. preliminary results are shown in Section 5 and we summarize and conclude in Section 6.

2 Inhomogeneous Diffusion

2.1 Isotropic Diffusion

Inhomogeneous diffusion started with the work of Perona and Malik [7]. In order to better situated this formalism we first discuss isotropic diffusion. In the isotropic case the filtering of the image is done via the solution of the isotropic diffusion equation

$$\begin{aligned} u_t &= c\Delta u = c\mathrm{div}(\nabla u) = \mathrm{div}(c\nabla u) \\ u(t=0) &= u_0 \quad . \end{aligned}$$

where Δu is the Laplacian of u, *div* is the divergent, ∇u is the gradient and c is a constant. This equation is called in image processing context linear scale-space because of its relation to convolution with a Gaussian with time dependent variance. The relation to linear filtering is done via the Green function (kernel) of this Partial Differential Equation (PDE):

$$\begin{aligned} u(x,y,t) &= \int G(x-x', y-y';\ t) u_0(x',y') dx'dy' \\ G(x,y;\ t) &= \frac{1}{4\pi t} e^{-\frac{x^2+y^2}{4t}} \quad . \end{aligned}$$

The relation of isotropic diffusion to the calculus of variations is given by the functional

$$S[u] = \int ||\nabla u(x,y)||_2^2 dxdy$$

and its gradient descent

$$u_t = \frac{\partial u}{\partial t} = -\frac{\delta S}{\delta u}$$

2.2 Inhomogeneous Diffusion

The idea of Perona and Malik was to use isotropic-like filtering far from the edges of the image and to reduce the smoothing near the edges in order to preserve the sharpness of the image. This was achieved in the PDE formulation via a local diffusion function

$$\begin{aligned} u_t &= \mathrm{div}(c(x,y)\nabla u) \\ u(t=0) &= u_0 \quad . \end{aligned}$$

The function $c(x,y)$ is the local diffusion coefficient and is usually taken as a monotonically decreasing function of $||\nabla u||_2$.

The relation to minimization of a functional was nicely formulated by the Φ-formalism of Deriche and Faugeras [5]. The functional they proposed is

$$S_\Phi[u] = \int \Phi\left(||\nabla u(x,y)||_2\right) dxdy \quad ,$$

and the gradient descent equation is

$$u_t = div\left(\frac{\Phi'(||\nabla u(x,y)||_2)}{||\nabla u(x,y)||_2}\nabla u\right)$$
$$u(t=0) = u_0 \quad .$$

The relation to Perona-Malik is given by $c(s) = \Phi'(s)/s$.

2.3 TV and MAP

Another approach that links functional minimization and inhomogeneous diffusion is Total Variation. In the original paper the functional is given by

$$S_{TV}[u] = \int \left[\frac{1}{2}\left(h * u(x,y) - u_0(x,y)\right)^2 + \lambda||\nabla u(x,y)||_2\right] dxdy \quad .$$

where h is a blur kernel and $*$ denotes convolution. The first term is called fidelity term and the second term is referred to as the smoothing term. The smoothing term is of the form of the Φ-formalism with $\Phi(s) = s$. The gradient descent equation reads

$$u_t = \lambda \text{div}\left(\frac{1}{||\nabla u(x,y)||_2}\nabla u\right) - \bar{h} * (h * u(x,y) - u_0(x,y))$$
$$u(t=0) = u_0 \quad , \quad \bar{h} = h(-x,-y) \ .$$

This equation is related to the Maximum A-posteriori Probability (MAP) method of statistical inference. Indeed by the Bayes rule the conditional probability of u given u_0 denoted by $P(u|u_0)$ is given by $P(u|u_0, h) \propto P(u_0|u, h)P(u)$ where $P(u)$ is the prior on the space of images. The relation to TV is given by

$$P(u_0|u,h) \propto \exp\{-\frac{1}{2}\int\left(h * u(x,y) - u_0(x,y)\right)^2 dxdy\}$$
$$P(u) \propto \exp\{-\lambda\int ||\nabla u(x,y)||_2 dxdy\}$$

The a-posteriori probability function is proportional to $\exp\{-S_{TV}\}$ and the MAP approximation is given by

$$\hat{u} = \arg\max_u P(u|u_0,h) = \arg\min_u S_{TV}[u] \quad .$$

We will refer to the relations between PDEs, functionals, filters and statistical inference in all the following analysis.

3 The Mumford-Shah Functional

The Mumford-Shah functional aims to simultaneously solve the problems of denoising and edge detection. For this end the functional is formulated as

$$S_{MS}[u,K] = \frac{1}{2}\int_\Omega (u-u_0)^2 \, dxdy + \lambda\int_{\Omega/K} ||\nabla u||_2^2 dxdy + \alpha\,(\text{length of } K) \quad ,$$

where the first term is the fidelity term. The second term dictates smoothing far from the set K of image discontinuities. The set K is assumed to be a set of continues curves and the last term is a penalty on the total length of these curves. The minimization and analysis of this functional are not simple since the set of discontinuities intervenes in the boundary of the integration. This is a free boundaries problem which is notoriously difficult. One of the best ways to deal with this problem is via the Γ-convergence technique, which was proposed by Ambrosio and Tortorelli.

3.1 The Ambrosio-Tortorelli Functional

In this approach one constructs a series (or a one-parametric family) of functionals that converge to a functional such that the limit of the series minimizers converges to the minimizer of the limit functional. The functionals in the series are easier to analyze. Ambrosio and Tortorelli suggested the following family of ϵ dependent functionals

$$S^{\epsilon}_{AT}[u,v] = \int_{\Omega} \left[\frac{1}{2}\left(u-u_0\right)^2 + \frac{\lambda}{2} v^2 ||\nabla u||_2^2 + \alpha \left(\epsilon ||\nabla v||_2^2 + \frac{(v-1)^2}{4\epsilon} \right) \right] dxdy$$

which Γ-converges to the Mumford-Shah functional when $\epsilon \to 0$. Here v is an auxiliary function that encodes the images discontinuity set: It approaches one in smooth regions and approach zero near an edge. The gradient descent for the image u leads to an inhomogeneous diffusion

$$\begin{aligned} u_t &= \lambda \mathrm{div}\left(v^2 \nabla u\right) - (u-u_0) \\ u(t=0) &= u_0 \quad . \end{aligned}$$

where the edge function v^2 plays the role of local diffusion coefficient.

The great difference from the TV and the Φ-formalism is in the way this diffusion coefficient is determined. In the latter methods the form of the diffusion coefficient is predefined. Here, in the Ambrosio-Tortorelli approach, this coefficient is a *dynamical variable* that is found by minimizing of the functional! The gradient descent equations read

$$\begin{aligned} v_t &= \alpha\epsilon\Delta v - \left(\lambda ||\nabla u||_2^2 v + \frac{\alpha(v-1)}{2\epsilon} \right) \\ v(t=0) &= 1 \quad . \end{aligned}$$

The advantage of dynamic determination of the edge set or equivalently the diffusion coefficient was shown in [3] for the case of de-blurring. The Ambrosio Tortorelli functional was shown to perform better than the TV and the Beltrami framework. The latter is the subject of the next section.

4 The Beltrami Framework

In this framework a two-dimensional (multi-channel) image is considered to be an imbedding of a surface in a higher dimensional manifold, or in more general terms an image is a section of the spatial-feature trivial bundle. The section is endowed with a metric and is, thus, a Riemannian manifold. The functional over the space of sections is the Polyakov action:

$$S_B[u,G] = \int \sum_{rs} (\nabla u^r)^T G^{-1} \nabla u^s H_{rs}(u) \sqrt{\det G}\, dxdy$$

where G is the metric of the image manifold, $H_{r,s}$ are entries of the metric of the spatial-feature space and the indices are for the different spatial and channel/feature, e.g. colors, of the image. One important parameter of the Beltrami framework is the ratio between spatial distance and feature distance. This ratio, termed here β is needed to measure distances in the combined spatial-feature space.

The inner product on the manifold is

$$< f,h >_G = \int f(x,y) h(x,y) \sqrt{\det G}\, dxdy$$

The EL equations of the functional with respect to this inner product necessitate division by $\sqrt{\det G}$. Assuming that H is the identity matrix the gradient descent equation for the image read

$$u^r_t = \frac{1}{\sqrt{\det G}} \mathrm{div} \left(\sqrt{\det G}\, G^{-1} \nabla u^r \right) \quad .$$

Note that G^{-1} plays the role of the diffusion tensor and leads to a true anisotropic diffusion flow. The metric is determined by minimizing the functional. The analytic solution is the induced metric.

5 The Beltrami-Mumford-Shah Functional

The main idea of this paper is to generalize the Mumford-Shah functional. Indeed, one can rewrite the MS functional via the AT approach as follows

$$S^\epsilon_{AT}[u,v] = \int_\Omega \left[\frac{1}{2}(u-u_0)^2 + \frac{\lambda}{2} (\nabla u)^T \begin{pmatrix} v^2 & 0 \\ 0 & v^2 \end{pmatrix} \nabla u + \alpha \left(\epsilon ||\nabla v||^2_2 + \frac{(v-1)^2}{4\epsilon} \right) \right] dxdy$$

This is a suggestive form that can be easily generalized to

$$S^\epsilon_{AT}[u,V] = \int_\Omega \left[\frac{1}{2}(u-u_0)^2 + \frac{\lambda}{2} (\nabla u)^T V^T V \nabla u + \alpha \left(\epsilon ||\nabla V||^2_F + \frac{||V-Id||^2_F}{4\epsilon} \right) \right] dxdy \ ,$$

where

$$V = \begin{pmatrix} v_{11} & v_{12} \\ v_{12} & v_{22} \end{pmatrix} \quad , \quad ||V||^2_F = v^2_{11} + 2v^2_{12} + v^2_{22} \quad .$$

and $||\nabla V||_F^2 = ||V_x||_F^2 + ||V_y||_F^2$. This turns the dynamic diffusion coefficient v into a *dynamic diffusion tensor* V!

The next observation is that we can write

$$G^{-1} = V^T V$$

and reinterpret the functional in a new way: Let u and u_0 be functions on a Riemannian manifold. We demand that the two functions be similar in the L_2 norm *on the manifold*. The metric is G and V^{-1} is the vielbein i.e. the symmetric square root of the metric. The fidelity and smoothness terms should be written on the manifold. The penalty term regards the metric (or the vielbein) only. It enforces it to be close to the identity matrix in smooth regions and drive the metric to a singular matrix that aligns along the discontinuity near an strong edge. The penalty regularizes the metric as well. The new functional, thus, read

$$S^{\epsilon}_{BMS}[u,V] = \int \left[\frac{1}{2}\left(u-u_0\right)^2 + \lambda\left(\nabla u\right)^T V^T V \nabla u\right] \frac{dxdy}{\det V} + \alpha \int \left(\epsilon ||\nabla V||_F^2 + \frac{||V - Id||_F^2}{4\epsilon}\right) dxdy \quad .$$

This functional generalizes the Mumford-Shah functional from a scalar diffusion coefficient to a tensor one going from inhomogeneous smoothing to a true anisotropic one. It also generalizes the Beltrami framework since the metric, that serves here as the diffusion tensor, is not predefined but is a dynamical variable that is fixed along the flow by the functional.

For multi-channel image, e.g. color image one may write

$$S^{\epsilon}_{BMS}[u,V] = \int \left[\frac{1}{2}\sum_r \left(u^r - u_0^r\right)^2 + \frac{\lambda}{2}\sum_{r,s}(\nabla u^r)^T V^T V \nabla u^s H_{r,s}(u)\right] \frac{dxdy}{\det V} + \alpha \int \left(\epsilon ||\nabla V||_F^2 + \frac{||V - Id||_F^2}{4\epsilon}\right) dxdy$$

where $H_{r,s}$ is the metric in the feature space e.g. color space.

The minimization equations, assuming that $H_{ab} = \delta_{ab}$, are

$$u_t^a = \lambda(\det V)\mathrm{div}\left(\frac{1}{\det V} V^T V \nabla u^a\right) - (u^r - u_0^r)$$

$$(V_{ij})_t = (V^{-1})_{ij}\sum_a (u^a - u_0^a) + \alpha\epsilon \Delta V_{ij} - \frac{\alpha(V-Id)_{ij}}{2\epsilon} - \frac{\lambda}{2}\sum_r (\nabla u^r)^T W_{ij} \nabla u^r$$

where

$$W_{11} = \begin{pmatrix} v_{11} & v_{12} \\ v_{12} & 0 \end{pmatrix} \quad , \quad W_{22} = \begin{pmatrix} 0 & v_{12} \\ v_{12} & v_{22} \end{pmatrix} \quad , \quad W_{12} = \begin{pmatrix} v_{12} & 2v_{11} \\ 2v_{22} & v_{12} \end{pmatrix} \quad ,$$

6 Results

6.1 Numerical Implementation

Let

$$\partial_x^f u := u(x+1,y) - u(x,y)$$

and

$$\partial_y^f u := u(x,y+1) - u(x,y)$$

be the forward finite difference approximation of $\partial_x(u)$ and $\partial_y(u)$ respectively. Similarly, backward derivatives are defined as

$$\partial_x^b u := u(x,y) - u(x-1,y)$$

and

$$\partial_y^b u := u(x,y) - u(x,y-1).$$

The forward gradient is therefore

$$\nabla^f(u) := (\partial_x^f, \partial_y^f)^T(u),$$

and the the backward gradient is given by

$$\nabla^b(u) := (\partial_x^b, \partial_y^b)^T(u).$$

Numerical scheme of the functional derivatives takes the form:

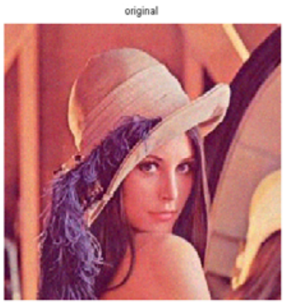
$$\delta\mathcal{F}_{,v_{11}} = -\frac{v_{22}\sum_c (u^c - u_0^c)^2}{2(v_{11}v_{22} - v_{12}^2)^2} + \lambda\left[v_{11}(\partial_x^f u)^2 + 2v_{12}\partial_x^f\partial_y^f u\right] + \frac{\alpha}{2\epsilon}(v_{11}-1) - 2\alpha\epsilon\nabla^b\cdot\nabla^f v_{11}.$$

$$\delta\mathcal{F}_{,v_{22}} = -\frac{v_{11}\sum_c (u^c - u_0^c)^2}{2(v_{11}v_{22} - v_{12}^2)^2} + \lambda\left[v_{22}(\partial_y^f u)^2 + 2v_{12}\partial_x^f\partial_y^f u\right] + \frac{\alpha}{2\epsilon}(v_{22}-1) - 2\alpha\epsilon\nabla^b\cdot\nabla^f v_{22}.$$

$$\begin{aligned}\delta\mathcal{F}_{,v_{12}} = &-\frac{v_{12}\sum_c (u^c - u_0^c)^2}{(v_{11}v_{22} - v_{12}^2)^2} \\ &+ \lambda\left[v_{12}(\partial_x^f u)^2 + v_{12}(\partial_y^f u)^2 + 2(v_{11}+v_{22})\partial_x^f u \partial_y^f u\right] + \frac{\alpha}{\epsilon}v_{12} - 4\alpha\epsilon\nabla^b\cdot\nabla^f v_{12}.\end{aligned}$$

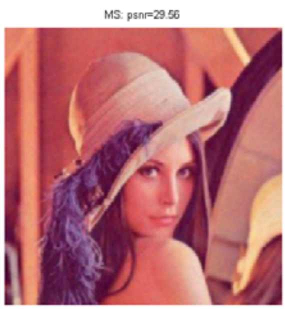
$$\delta\mathcal{F}_{,u^r} = (u^r - u_0^r) - \lambda \det V \nabla^b\left(\frac{1}{\det V}V^T V \nabla^f u^c\right)$$

Optimization was carried out using the alternate minimization technique using the line search strategy. Descent direction was computed as gradient descent, and step size was calculated by Armijo rule [2]. The algorithm stops whenever all variables have reached convergence tolerance ε. The algorithm of Tensor-MS method is given below.

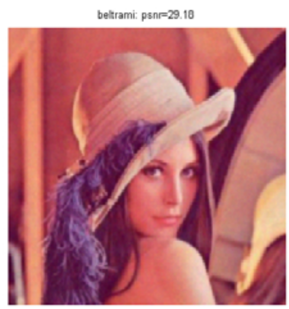


(a) Original image

(b) Beltrami-Mumford-Shah flow. PSNR= 30.21



(c) Mumford-Shah flow. PSNR= 29.56



(d) Beltrami flow. PSNR= 29.13

Algorithm Energy Descent(u_0)

- Initialize $u^0 = u_0, v_{11}^0 = 1, v_{22}^0 = 1, v_{12}^0 = 1, k = 0$
- Do
  1. $\tau^{v_{11}} = \text{ArmijoStep}(\mathcal{F}, v_{11}^k)$
  2. $v_{11}^{k+1} = v_{11}^k - \tau^{v_{11}} \delta\mathcal{F}_{,v_{11}}(v_{11}^k, v_{22}^k, v_{12}^k, u^k)$
  3. $\tau^{v_{22}} = \text{ArmijoStep}(\mathcal{F}, v_{22}^k)$
  4. $v_{22}^{k+1} = v_{22}^k - \tau^{v_{22}} \delta\mathcal{F}, v_{22}(v_{11}^{k+1}, v_{22}^k, v_{12}^k, u^k)$
  5. $\tau^{v_{12}} = \text{ArmijoStep}(\mathcal{F}, v_{12}^k)$
  6. $v_{12}^{k+1} = v_{12}^k - \tau^{v_{12}} \delta\mathcal{F}, v_{12}(v_{11}^{k+1}, v_{22}^{k+1}, v_{12}^k, u^k)$
  7. $\tau^u = \text{ArmijoStep}(\mathcal{F}, u^k)$
  8. $u^{k+1} = v_{12}^k - \tau^u \delta\mathcal{F}_{,u}(v_{11}^{k+1}, v_{22}^{k+1}, v_{12}^{k+1}, u^k)$
- while $\|v_{11}^{k+1} - v_{11}^k\|, \|v_{22}^{k+1} - v_{22}^k\|, \|v_{12}^{k+1} - v_{12}^k\|, \|u^{k+1} - u^k\| \geq \varepsilon$

Beltrami and MS methods are similarly implemented using the corresponding derivatives. Parameter set for two images are given in the following table, where in all cases tolerance was set to $\varepsilon = 10^{-3}$.

	Tensor MS			MS			Beltrami	
	α	λ	ϵ	α	λ	ϵ	λ	β
Ballet	0.1	0.5	0.01	0.1	0.6	0.01	0.65	1.0
Lenna	0.1	0.5	0.1	0.1	0.6	0.01	0.68	1.0

6.2 Results

(e) Original image

(f) Beltrami-Mumford-Shah flow. PSNR= 32.57

(g) Mumford-Shah flow. PSNR= 31.99

(h) Beltrami flow. PSNR= 31.83

7 Summary and Conclusions

We present in this paper a unifying generalization of the Mumford-Shah functional and the Beltrami framework. We show that we are able, in this way, to have the benefits of true anisotropic diffusion together with a dynamically tuned metric/diffusion tensor. The functional is naturally defined in terms of the vielbein-the metric's square root. preliminary results show improvement on both the Beltrami flow and the Mumford-Shah flow.

References

1. Ambrosio, L., Tortorelli, V.M.: Approximation of functionals depending on jumps by elliptic functionals via Γ-convergence. Comm. Pure Appl. Math. 43, 999–1036 (1990)
2. Armijo, L.: Minimization of functions having Lipschitz continuous first partial derivatives. Pacific J. Math. 16(1), 13 (1966)
3. Bar, L., Sochen, N., Kiryati, N.: Image Deblurring in the Presence of Impulsive Noise. International Journal of Computer Vision 70(3), 279–298 (2006)
4. Bar, L., Sochen, N., Kiryati, N.: Semi-Blind Image Restoration via Mumford-Shah Regularization. IEEE Trans. on Image Processing 15(2), 483–493 (2006)
5. Deriche, R., Faugeras, O.: Les EDP en Traitement des Images et Vision par Ordinateur. Traitement du Signal 13(6) (1996)
6. Mumford, D., Shah, J.: Optimal Approximations by Piecewise Smooth Functions and Associated Variational Problems. Comm. on Pure and Applied Math. XLII(5), 577–684 (1989)
7. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence 12(7), 629–639 (1990)
8. Rosenau, P.: Free energy functionals at the high-gradient limit. Phys. Rev. A (Rapid Communications) 41(4), 2227–2230 (1990)
9. Rudin, L., Osher, S., Fatemi, E.: Nonlinear Total Variation based noise removal algorithms. Physica D 60, 259–268 (1992)
10. Sochen, N., Kimmel, R., Malladi, R.: A general framework for low level vision. IEEE Transactions on Image Processing 7, 310–318 (1998)
11. Weickert, J.: Coherence-Enhancing Diffusion Filtering. International Journal of Computer Vision 31(2/3), 111–127 (1999)