

Fast image processing with constraints by solving linear PDEs

Witali Kusnezow*, Wilfried Horn⁺, and Rolf P. Würtz⁺

* *Teleca Systems GmbH, Rensingstr. 55, D-44807 Bochum, Germany*

⁺ *Institut für Neuroinformatik, Ruhr-University, D-44780 Bochum, Germany*

Received 20 November 2006; revised 29 May 2007; accepted 28 December 2007

Abstract

We present a general framework that allows image filtering by minimization of a functional using a linear and positive definite partial differential equation (PDE) while also permitting to control the weight of each pixel individually. Linearity and positive definiteness allow to use fast algorithms to calculate the solution. Pixel weighting allows to enforce the preservation of edge information without the need for nonlinear diffusion by making use of information coming from an external source. The proof of existence and uniqueness of the solution is outlined and based on that a numerical scheme for finding the solution is introduced.

Using this framework we developed two applications. The first is simple and fast denoising, which incorporates an edge detection algorithm. In this case the functional is designed to enhance the weight of the approximation term over the smoothing term at those places where an edge is detected.

The second application is a background suppression algorithm that is robust against noise, shadows thrown by the object, and on the background and varying illumination.

The results are qualitatively not quite as good as the ones obtained with nonlinear PDEs, but this disadvantage is compensated by the processing speed, which allows analysis of a 320 by 240 pixel color frame in about 0.3s on a standard PC.

1 Introduction

Preprocessing of images has always been an important topic in computer vision since most of the data that is collected suffers from blur, interpolation errors, color, geometric, refractive or other distortions, and camera noise. Methods for noise reduction include linear, median, Wiener and morphological filters as well as wavelet based denoising [15, 12, 16, 17, 14, 6, 8, 10, 1]. Partial differential equations (PDEs) have proven to be a useful tool in that area. Early works [19] using simple linear PDEs suffer from blurring edges, and more complex non-convex energy functionals [12, 2, 5, 11, 7], lead to nonlinear PDEs which converge slowly, even in the few cases for which convergence can be proved.

This paper presents a method using a PDE that is both linear and able to incorporate external information to weight each image pixel individually. This allows proving the existence of the solution and to compute it fast enough to get close to online preprocessing of image sequences. The problem of edge blurring is tackled by using an edge detector whose output is used as the external information that is incorporated into the functional. This allows to flexibly use different edge models. The computation time for the edge detector is negligible compared to that for the solution of the PDE.

Correspondence to: <rolf.wuertz@neuroinformatik.rub.de>

Recommended for acceptance by Jon Sporring

ELCVIA ISSN:1577-5097

Published by Computer Vision Center / Universitat Autònoma de Barcelona, Barcelona, Spain

We show that this framework is useful for background suppression, since no pixelwise decision is made, but whole areas are taken into account so that the problem of single pixels with a wrong classification can be reduced immensely.

The paper is organized as follows. In the following section we describe the mathematical problem formulation, sketch the proofs for existence and uniqueness of the solution, construct a linear operator to calculate it, and briefly present a fast numerical implementation. In Section 3 we apply the method to two useful applications. In section 4 we present the results of noise reduction of artificial as well as natural images and of image segmentation using our general framework in both RGB and the CIE Lab color spaces. We compare results from both applications with work that has already been done in these fields. In the final section we draw conclusions and present an outlook.

2 Theory

In this section we will present the general functional that we use throughout this paper. Based on ideas presented in [9], we prove the existence of a solution of this functional as well as its properties. The proofs of the lemmata are given in the appendix.

2.1 Base energy functional

Consider the following energy functional on a set of finite measure $\Omega \subset \mathbb{R}^2$

$$F := \frac{1}{2} \int_{\Omega} k(\vec{x})(u - f)^2 d\vec{x} + \frac{1}{2} \int_{\Omega} p(\vec{x}) (\nabla u)^2 d\vec{x}, \quad (1)$$

with k, p smooth, positive and bounded functions, $0 < k_0 \leq k(\vec{x}) \leq k_1 < \infty$, $0 < p_0 \leq p(\vec{x}) \leq p_1 < \infty$, f being a given input signal and u being the function varied to minimize the functional.

The function $k(\vec{x})$ allows us to give the approximating part of the functional different weights at different places. This enables us to give that term a very high weight compared to the smoothing term depending on the local circumstances, e.g., where edges are present.

The first variation of the energy functional is

$$F(u + \delta) - F(u) \approx \delta F = \frac{1}{2} \int_{\Omega} k u \delta d\vec{x} + \frac{1}{2} \int_{\Omega} p \nabla u \cdot \nabla \delta d\vec{x} - \frac{1}{2} \int_{\Omega} k f \delta d\vec{x}, \quad (2)$$

which is zero at the minimum of F for all δ . Defining the scalar products

$$\langle u, v \rangle_{\tilde{\mathbb{L}}_2} := \int_{\Omega} k u v d\vec{x}, \quad (3)$$

$$\langle u, v \rangle_{\tilde{\mathbb{H}}^1} := \int_{\Omega} k u v d\vec{x} + \int_{\Omega} p \nabla u \cdot \nabla v d\vec{x}, \quad (4)$$

and the Hilbert spaces $\tilde{\mathbb{L}}_2(\Omega)$ and $\tilde{\mathbb{H}}^1(\Omega)$ as all functions with finite respective norm allows to rewrite (2) as

$$\langle u, \delta \rangle_{\tilde{\mathbb{H}}^1} = \langle f, \delta \rangle_{\tilde{\mathbb{L}}_2}. \quad (5)$$

This equation is valid for all functions $\delta \in \tilde{\mathbb{H}}^1$.

The following lemma is an intermediate step to the proof that a unique solution to (5) exists for each function $f \in \tilde{\mathbb{L}}_2$. The inequality can be derived from Poincaré's inequality.

Lemma 1 1. Each element $u \in \tilde{\mathbb{H}}^1$ belongs to $\tilde{\mathbb{L}}_2$ $\tilde{\mathbb{H}}^1 \subset \tilde{\mathbb{L}}_2$;

2. The linear subspace $\tilde{\mathbb{H}}^1$ is dense in $\tilde{\mathbb{L}}_2$ $\overline{\tilde{\mathbb{H}}^1} = \tilde{\mathbb{L}}_2$;

$$3. \exists a > 0 \forall u : \|u\|_{\tilde{\mathbb{L}}_2} \leq a \|u\|_{\tilde{\mathbb{H}}^1}$$

Existence and the uniqueness of the solution to equation (5) follows from the following lemma, which is a consequence of the Riesz representation theorem.

Lemma 2 *If E, F are Hilbert spaces, F is a dense subset of E : $F \subset E, \overline{F} = E$ and*

$$\exists a > 0 \forall x \in F \quad \|x\|_E \leq a \|x\|_F. \quad (6)$$

Then the equation

$$\forall \delta \in F \quad \langle x, \delta \rangle_F = \langle f, \delta \rangle_E \quad (7)$$

has an unique solution for each $f \in E$.

The linear operator V that solves equation (5):

$$u = Vf \quad f \in \tilde{\mathbb{L}}_2, \quad u \in \tilde{\mathbb{H}}^1 \quad (8)$$

acts in the way that

$$\langle x, Vy \rangle_{\tilde{\mathbb{H}}^1} = \langle x, y \rangle_{\tilde{\mathbb{L}}_2} \quad (9)$$

and has the following properties:

Lemma 3 1. $V : \tilde{\mathbb{L}}_2 \rightarrow \tilde{\mathbb{H}}^1$ is a bounded linear operator and $\|V\| \leq a$;

2. $V : \tilde{\mathbb{L}}_2 \rightarrow \tilde{\mathbb{L}}_2$ is self-adjoint;

3. $V : \tilde{\mathbb{H}}^1 \rightarrow \tilde{\mathbb{H}}^1$ is a positive operator;

Lemma 2 states the existence of a solution, but it does not enable us to calculate it. To find the solution we need to consider the elliptical inverse operator of V : $A := V^{-1}$ which is defined on the dense subspace $D(A) := V\tilde{\mathbb{L}}_2 = \{Vu \mid u \in \tilde{\mathbb{L}}_2\}$. Using A the integral equation (5) is given by

$$Au = f; \quad f \in \tilde{\mathbb{L}}_2, \quad u \in D(A) \quad (10)$$

Lemma 4 1. A is a positive definite operator in $\tilde{\mathbb{L}}_2$:

$$A = A^*, \quad \exists \gamma > 0 \forall x \in D(A) \quad \langle Ax, x \rangle_{\tilde{\mathbb{L}}_2} \geq \gamma \|x\|_{\tilde{\mathbb{L}}_2}^2$$

2. If $u \in C^2(\Omega)$, equation (5) can be written as an elliptical partial differential equation with the Neumann boundary conditions:

$$ku - \operatorname{div}(p\nabla u) = kf \quad (\Omega), \quad (11)$$

$$\frac{\partial u}{\partial n} = 0 \quad (\partial\Omega). \quad (12)$$

This can be shown using Green's formula.

To find a solution of the Neumann boundary problem of (11) either Gauss-Seidel or SOR-methods are often used. We have shown that the operator A is positive definite and this knowledge can be used to significantly enhance the performance of the PDE solver.

Problem (1) can be generalized to vector-valued functions $\vec{u}$ with components u_i :

$$F := \frac{1}{2} \int_{\Omega} \sum_{i=1}^n k_i(\vec{x})(u_i - f_i)^2 d\vec{x} + \frac{1}{2} \int_{\Omega} \sum_{i=1}^n p(x)(\nabla u_i)^2 d\vec{x}. \quad (13)$$

We require that k_1 be positive and bounded, while for $i \geq 2$ k_i may also be zero. Then the conditions of lemma 2 are fulfilled and we find

Lemma 5 *The norm introduced by the energy given in (13):*

$$\|u\|_s^2 = \int \sum_{i=1}^n k_i(\vec{x}) u_i^2 d\vec{x} + p(\vec{x}) \int \sum_{i=1}^n (\nabla u_i)^2 d\vec{x} \quad (14)$$

is equivalent to the norm introduced by (1).

Also from lemma 2 the existence of an unique solution is given and the operator V even has the properties denoted in lemma 3, and its inverse operator $A := V^{-1}$ is also positive definite. In addition, if $u \in C^2$, it also solves the Euler equation:

$$\sum_{i=1}^n k_i u_i - \operatorname{div}(p \nabla u_i) = \sum_{i=1}^n k_i f_i \quad (\Omega), \quad (15)$$

$$\frac{\partial u_i}{\partial n} = 0 \quad (\partial\Omega). \quad (16)$$

2.2 Numerical algorithm

To develop a numerical algorithm to solve (15) and consequently minimize (1) we now consider an unknown function $u(x, y)$ and a known function $f(x, y)$ defined on a rectangular area of pixels with dimensions X and Y .

The Laplace operator, which appears in (11) and (15) if $p(\vec{x}) = \text{const}$, can be approximated by:

$$\Delta u(x, y) \approx u(x \pm 1, y) + u(x, y \pm 1) - 4u(x, y) =: D_i u(x, y); \quad (17)$$

and the anisotropic diffusion operator for $p(\vec{x}) \neq \text{const}$ is approximated by

$$\operatorname{div}(p(\vec{x}) \nabla u(\vec{x})) \approx \sum_{\vec{\chi} \in \mathcal{N}(\vec{x})} \frac{1}{2} (p(\vec{\chi}) + p(\vec{x})) (u(\vec{\chi}) - u(\vec{x})) =: D_a(\vec{x}), \quad (18)$$

where $\mathcal{N}(\vec{x})$ is the set of all four neighbors of the pixel at $\vec{x}$. To satisfy the Neumann boundary condition we assume that all $\vec{x}$ with $x_1 \in \{-1, X+1\}$ or $x_2 \in \{-1, Y+1\}$ are equal to their nearest neighbors to avoid special cases at the image boundaries.

The anisotropic elliptical differential equation (11) with boundary conditions (12) is approximated by a linear equation and the discretized operator is called A_a :

$$A_a u := (k - D_a)u = kf, \quad \{u, f\} \subset \mathbb{R}^{X \cdot Y}. \quad (19)$$

The generalization to vector-valued functions is straightforward, so eqs. (15) and (16) are solved by solving

$$A_a \vec{u} := (\vec{k} - D_a) \vec{u} = \vec{k} f, \quad \{\vec{u}, f\} \subset \mathbb{R}^{X \cdot Y \cdot n}. \quad (20)$$

The operator in this approximation is a linear and positive definite on $\mathbb{R}^{X \cdot Y}$. Equation (19) can be solved iteratively because the matrix of the operator is sparse. The positivity allows to use the conjugated gradients method. It produces a chain of vectors $x_i \in \mathbb{R}^{X \cdot Y}$, which terminates after at most $X \cdot Y$ steps at the desired solution $x_{X \cdot Y}$. In each step $x_k \rightarrow x_{k+1}$ a k -dimensional minimization is carried out:

$$F(x_{k+1}) = \min_{u_0, \dots, u_k} F(x_0 + u_0 r_0 + \dots + u_k r_k), \quad (21)$$

$$r_i := f - A x_i \quad \text{for } i \leq k. \quad (22)$$

Preconditioning can be used to increase the speed of convergence of this method using a simple inverse positive definite matrix B , so that $B^{-1}A$ is good approximation of the unit matrix. A complete description of those and a comparison with other iterative methods can be found, e.g., in [18].

3 Applications

3.1 Integration of edge information in the variational problem

There are two straightforward ways to integrate edge information into a variational formulation with only a small performance penalty. First we introduce a smooth function χ_C that is one in an ε -area around edges in the continuum and falls to zero steeply but in a smooth fashion.

The first method is the minimization of the *isotropic* energy functional

$$F_i := \frac{1}{2} \int_{\Omega} (1 + M_c \chi_c)(\vec{x}) (u - f)^2 d\vec{x} + \frac{1}{2} \lambda \int_{\Omega} (\nabla u)^2 d\vec{x}, \quad (23)$$

with $M_c \gg 0$ being a large positive constant and the positive constant λ replacing the more general function $p(\vec{x})$. The large value M_c ensures that points on edges receive a much higher weight than points that are not edges, which results in the effect that the smoothing term becomes less important and edges are approximated much closer to the original value of the pixel at that position. Note that $(1 + M_c \chi_c)(\vec{x}) = k(\vec{x})$ in (2) stays constant in the sense that is not altered during the variation of u . It only incorporates the external information coming from an edge detector in this case. Any linear or nonlinear system for detecting edges can be used. During the iteration process $k(\vec{x})$ is not changed in any way.

The second method is a minimization of the *anisotropic* energy term

$$F_a := \frac{1}{2} \int_{\Omega} (u - f)^2 d\vec{x} + \frac{1}{2} \lambda \int_{\Omega} \frac{(\nabla u)^2}{(1 + M_c \chi_c)(\vec{x})} d\vec{x}. \quad (24)$$

Here $p(\vec{x}) = \frac{1}{(1 + M_c \chi_c)(\vec{x})}$ is also not varied during the optimization of u . Both functionals are of form (1), which means that we can use the method described.

The second functional (24) results in the stationary anisotropic diffusion equation

$$u - \lambda \operatorname{div} \left(\frac{\nabla u}{1 + M_c \chi_c} \right) = f \quad (\Omega), \quad (25)$$

whereas (23) results in a partial differential equation like the stationary Schrödinger equation:

$$(1 + M_c \chi_c)u - \lambda \Delta u = (1 + M_c \chi_c)f \quad (\Omega). \quad (26)$$

which can be converted to the isotropic diffusion:

$$u - \frac{\lambda}{1 + M_c \chi_c} \Delta u = f \quad (\Omega). \quad (27)$$

The parameter M_c helps to approximate edges more accurately than other points. Note that the edges in this model are not represented by a curve but by a compact subset $C \subset \Omega$ with a positive Lebesgue measure.

3.2 Image smoothing with respect to edges

As an application of the proposed scheme we consider the well known problem of noise filtering of an image with accurate edge processing. For the edge detection we use the approach proposed in [3]. For gray level images, we can use the functionals (23) or (24), and a generalization for color images (n channels) is easily given by using:

$$E_i(u, c) = \frac{1}{2} \int_{\Omega} (1 + M_c \chi_c) \sum_{i=1}^n (f_i - u_i)^2 d\vec{x} + \frac{1}{2} \lambda_u \int_{\Omega} \sum_{i=1}^n (\nabla u_i)^2 d\vec{x}, \quad (28)$$

$$E_a(u, c) = \frac{1}{2} \int_{\Omega} \sum_{i=1}^n (f_i - u_i)^2 d\vec{x} + \frac{1}{2} \lambda_u \int_{\Omega} \sum_{i=1}^n \frac{(\nabla u_i)^2}{(1 + M_c \chi_c)} d\vec{x}. \quad (29)$$

1. Initialization

- (a) Calculate the edges of the background image (χ_{c_0}).
- (b) Filter the image with respect to edges (if necessary).
- (c) Store the background and its edge data.

2. Sequence processing:

- (a) Calculate the edges of the current picture using the same edge detector with the same parameters as in 1a.
- (b) Smooth the picture with respect to the edges (if necessary).
- (c) Calculate the overlap and the symmetric difference of edges found for the background and the current image:

$$\chi_{\cap} := \chi_{c_0} \chi_{c_N}; \quad \chi_{\Delta} := \max\{\chi_{c_N}, \chi_{c_0}\} - \chi_{\cap}; \quad (30)$$

The first one being an AND operation whereas the second is an XOR operation.

- (d) Calculate the probability $u(\vec{x}) \in [0, 1]$ for each point $\vec{x}$ to belong to the object by solving the Euler equation (33) by gradient descent as described in section 2.2.
- (e) Separate the foreground from the background using a threshold $\theta \in (0, 1)$. Then the **Object** contains all points $\vec{x}$ with $u(\vec{x}) > \theta$, and the **Background** contains all other points.

Figure 1: Algorithm for segmentation. The letter u is used for the probability to adhere to the nomenclature.

for the isotropic (28) and the anisotropic (29) case. Minimization of this functional leads to the solution of problem (11) for each channel.

To test the robustness of the proposed schemes against noise we added Gaussian noise with amplitude α and variance σ^2 to the images, calculated their edges, filtered them and compared the result to the original by the signal-to-noise ratio (SNR). For a numerical comparison we used a signal-to-noise ratio given by $10 \log_{10} (\|f - u\|^2 / \|f\|^2)$. For all tests we used $\sigma = 7$ and different values of A . Even better results could be gained by repeating the method twice, i.e., using the result of this procedure for yet another edge extraction. The edges from this second step were a lot better than those from the noisy one. Then we used the edges obtained in the second step to filter the noisy image to get a real clean image. The number of one repetition of the combination of edge extraction and smoothing is empirical and has been fixed as a good compromise between perceived edge quality and computation time.

3.3 Image Segmentation

Several problems arise for a background suppression segmentation of camera sequences. General problems include camera noise and variations in lighting. Segmentation specific, the object may contain colors from the background or it may throw a shadow on the background resulting in different colors on the background itself. The usage of our variational method together with edge information allows to enhance the quality of object discrimination. This is done by calculating a function that indicates whether a pixel or an area belongs to the background or to an object an smoothing that function using the same functional we used for smoothing an image.

The variational scheme proved to be robust against the noise and small lighting variations. To overcome the other problems we needed to use additional information about the sequence. We used an additional cue based

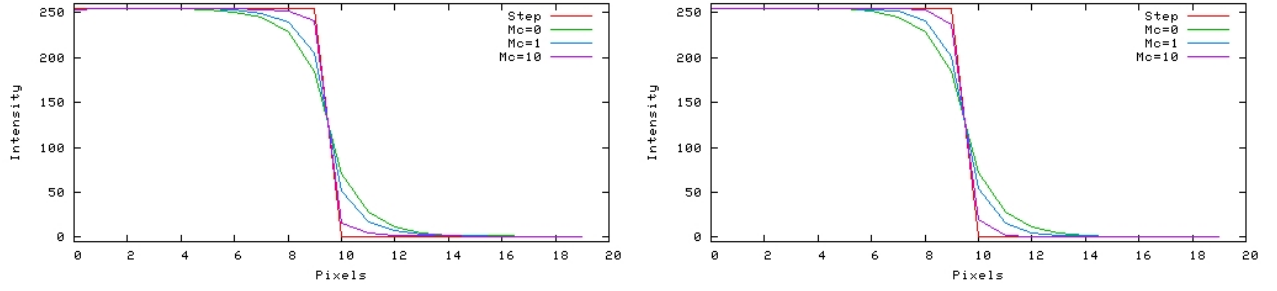


Figure 2: Anisotropic (left) and isotropic (right) edge smoothing for $\lambda = 1$.

on edge information generated by using the following assumptions:

1. If at a given point an edge is located in the picture with the object and without the object, that point probably belongs to the background.
2. If at a given point an edge is presented on background image, but not on the picture with the object (or vice versa), that point probably belongs to the object.

Weighting rules 1 and 2 with constants $M_{\cap} \gg 1$ and $M_{\Delta} \gg 1$, which serve the same purposes as M_c in 3.1, the expected object size with β , and the smoothness of the indicator function with λ_p yields the functional

$$F_s(u) = \frac{1}{2} \int_{\Omega} (\beta + M_{\cap} \chi_{\cap}) (1 - u)^2 + (\|I_S - B_S\|_{\mathbb{R}^3}^2 + M_{\Delta} \chi_{\Delta}) u^2 d\vec{x} + \frac{1}{2} \lambda_p \int_{\Omega} (\nabla u)^2 d\vec{x}. \quad (31)$$

Note that u does not represent the image itself anymore. The meaning of values of u is the distinction between a pixel or an area very likely to belong to the background (which is the case for $u(\vec{x}) = 0$), very likely to belong to the object (which is the case for $u(\vec{x}) = 1$) or something in between.

By minimizing this we calculate an indicator function that describes the likelihood for each pixel to belong to the background and use our approach to smooth that indicator function. B_S is the smoothed background image which is the result of (28) or (29) for the background image, whereas I_S is the smoothed object image which is the result of the same optimization for the object image. The algorithm is shown in fig.1

Note that (31) means that the likelihood is described as a function over the whole image. The likelihood itself is 'smoothed' so that the value of the indicator function for a specific pixel to belong to the object depends on its neighbors. This yields a more region-based segmentation so that no morphological operation is necessary after the segmentation.

The energy functional (31) is of the kind (13) and its Euler equation

$$(\beta + M_{\cap} \chi_{\cap} + \|I_S - B_S\|^2 + M_{\Delta} \chi_{\Delta}) u - \lambda_p \Delta u = \beta + M_{\cap} \chi_{\cap} \quad (\Omega) \quad (32)$$

$$\frac{\partial u}{\partial n} = 0 \quad (\partial\Omega) \quad (33)$$

is solved using the above proposed scheme.

4 Results

4.1 Image filtering

The influence of edge information on smoothing can be demonstrated by the profiles in fig. 2. One can see that both schemes are able to approximate edges very well. The noisy images are shown in figs. 3 a) and 4 a), the

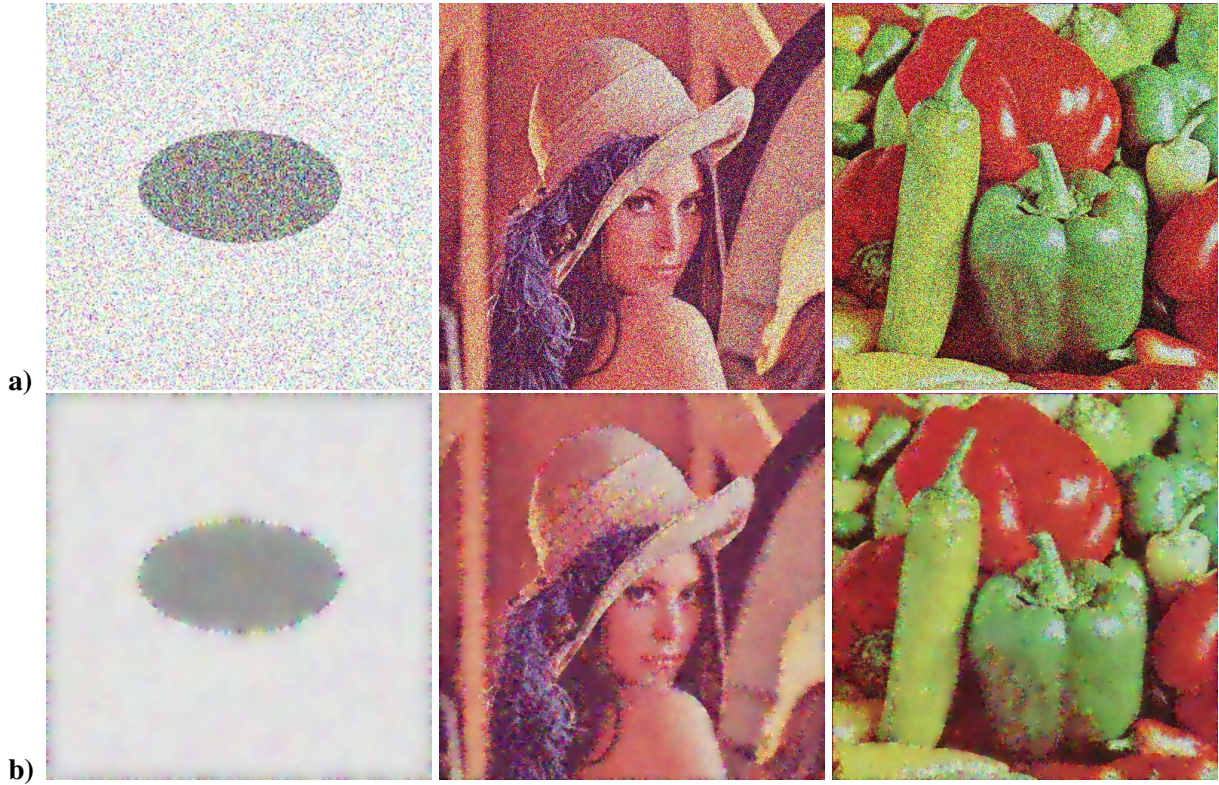


Figure 3: Results of smoothing. **a):** Images with artificial noise with an amplitude of 10. The SNRs from left to right are 13.48, 6.74 and 6.08. **b):** Results of image filtering using the isotropic scheme. The resulting SNRs are 17.7986, 15.4621 and 14.2257.

smoothing results using the anisotropic and the isotropic scheme can be seen in fig. 3 b) and 4 c), respectively. Fig. 4 also shows the intermediate edges and the result of the iterated smoothing. The isotropic scheme is not really suitable for very noisy images, but the anisotropic scheme yields very good results, especially if iterated twice. This can be examined in the close-up in fig. 5. More than two steps did not enhance the result anymore.

Since it may be argued that Gaussian noise is a poor approximation to reality, we also tested our algorithm with an image from [20] that suffers from real camera noise. In fig. 6 we compare the results from [20] with ours. Their result is slightly better than ours but at the expense of having to solve a nonlinear PDE, which leads to a required computation time of more than a minute on an Athlon XP2800+.

Our algorithm on the same machine needs about 0.01s per iteration per channel. For each channel, 5-20 iterations are required to yield an approximation of $\|Au - f\|^2 \leq 0.5^2 \cdot X \cdot Y$ with $u, f \in [0, 255]$. For typical values of λ ten or even less iterations are needed. Altogether, for a three channel color image of size $320 \cdot 240$ we roughly need 0.3s to compute the solution of the functional, and never encountered calculation times over 0.5s. This could certainly be reduced further by code optimization.

4.2 Segmentation

In fig. 7 and the accompanying video clips we present our results for a segmentation using the method described in section 3.3. On the left of fig. 7 a) (video1.mpg, video1-segmented.mpg) one can see a fairly easy task for a segmentation process, since the colors of the person to be segmented are mostly rather different from the background, and the result is good. But our algorithm can also deal with rather hard examples as one can see on the left image of fig. 7 b) (video2.mpg). First, the person to be segmented wears a shirt with background color. Second, there is heavy shadow on the white wall, which causes problems to a purely color-difference-

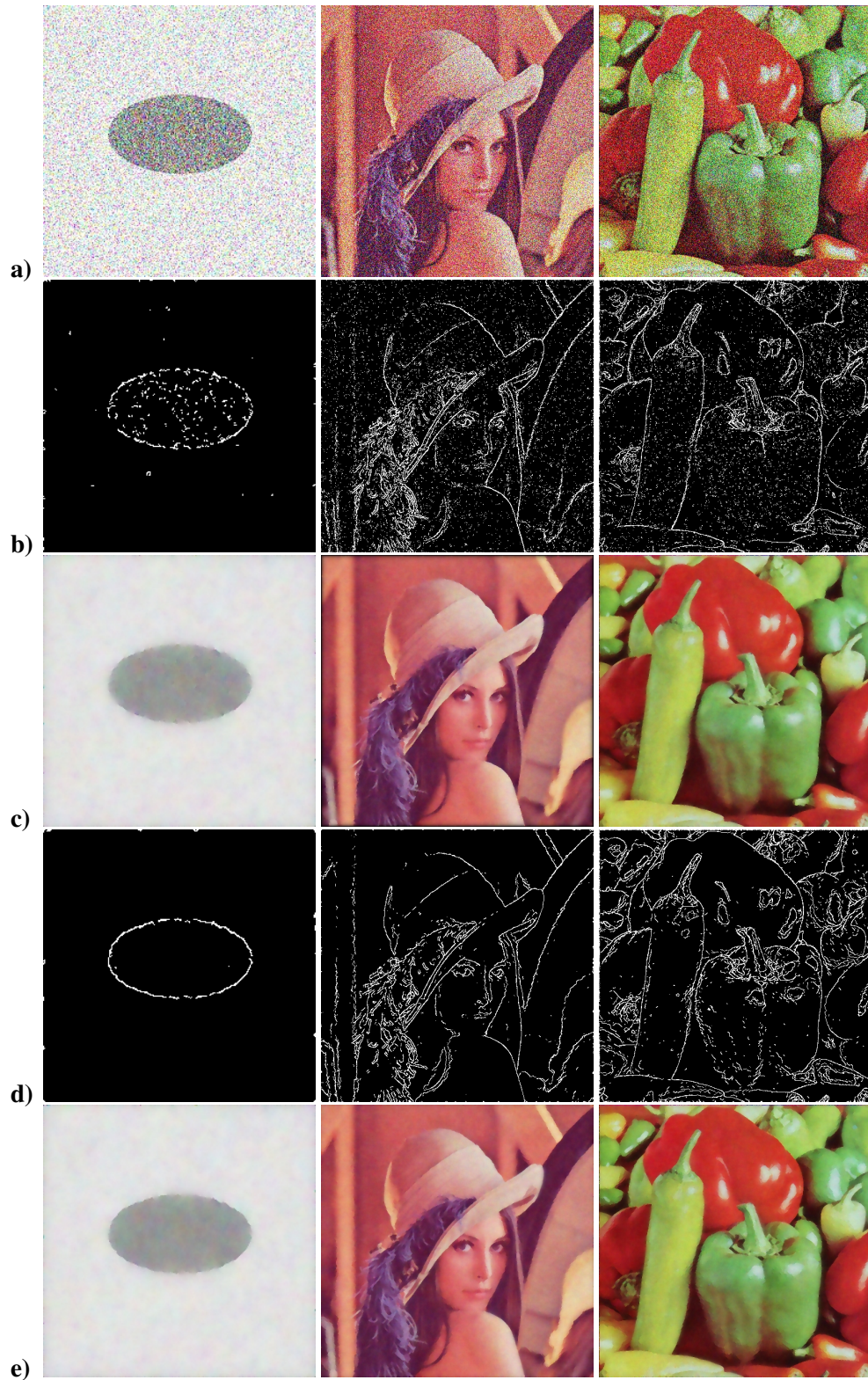


Figure 4: Results of smoothing. **a)**: Images with artificial noise with an amplitude of 10. The SNRs from left to right are 13.48, 6.74 and 6.08. **b)**: Edges extracted from the noisy images with lots of artifacts. **c)**: Result of anisotropic smoothing using the edges from b). **d)**: Edges extracted from the smoothed images in c). **e)**: Result of anisotropic smoothing using the edges from d); the SNRs are 18.68, 18.89 and 16.94.

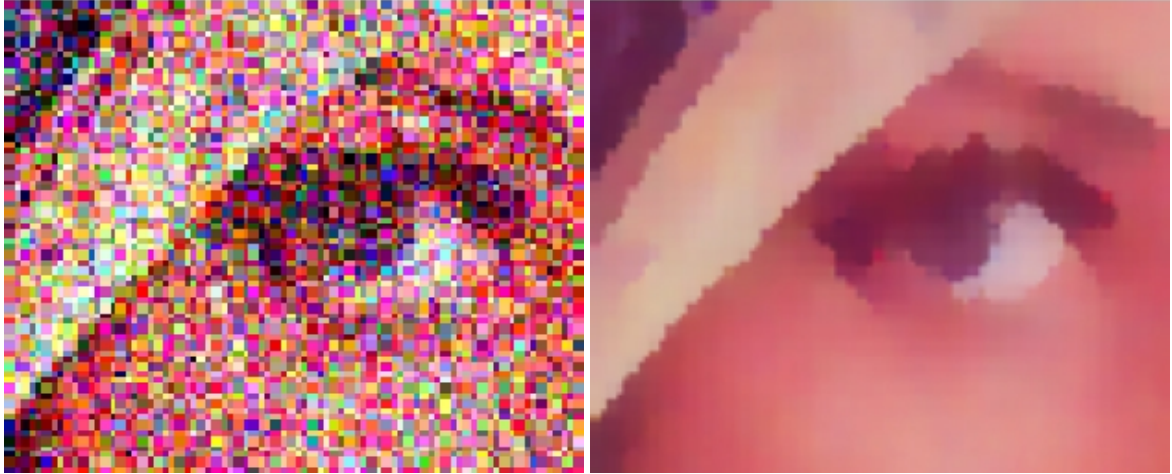


Figure 5: Close up smoothing results for the anisotropic scheme. On the left a magnification of the noisy image of Lenna from fig. 4 a), on the right the result from anisotropic filtering.

based algorithm. Third, the illumination condition is hard, since there is a strong light source on the left, so the same objects have different colors depending on surface orientation relative to the light source. Fourth, the fast motion already blurs the base image. Nevertheless the segmentation shows a useful result as can be seen in the center image of fig. 7 b) (video2-segmented.mpg).

As with any other low level segmentation there are also images where our approach fails. Examples of such a sequence are given in fig. 7 c). This failure is the result of several bad circumstances in that sequence. First, the white board in the center of the image is frontally illuminated strongly from the front and it reflects that light so that it is very bright. The shadow coming from the person on the right completely removes that reflection on the shadowed areas and the color difference due to shadow is a lot bigger then usual since the reflection enhances the effect a lot. This shadow is too strong for the method to deal with and therefore the lower right corner of the white board appears in the segmented image. The second reason for the failure is the low color difference between the sweater worn by the man on the right and the background. Additionally, both background and sweater are without texture so that there are neither edges on the image to be segmented nor on the background. That is the reason why the sweater is not segmented correctly. The reason for the rather large margins around the segmented objects is a rather large value for λ that was necessary for other parts of the image. These are all circumstances which are problematic for our approach and the given sequence already reflects all of them.

The time needed for segmentation depends on the task and ranges between 0.5 and 1 seconds. We used the L^*a^*b color space since this leads to slightly better results than RGB.

These results have been obtained without using any other cues (for example movement detection or model knowledge). Additional cues can be easily incorporated by converting them into coefficients for $(1 - p)^2$ or p^2 in (31). Depending on the complexity this would of course raise the computing time, since those additional cues would have to be calculated as well. However, the time needed for computing results of additional low level cues can be reduced by using parallel computing. Since low level cues can often be computed independently from each other, almost no additional time would be needed.

The computation time for the algorithm would be almost unaffected since only some addition operations would have to be added, but this procedure would also probably increase the reliability of the segmentation result.

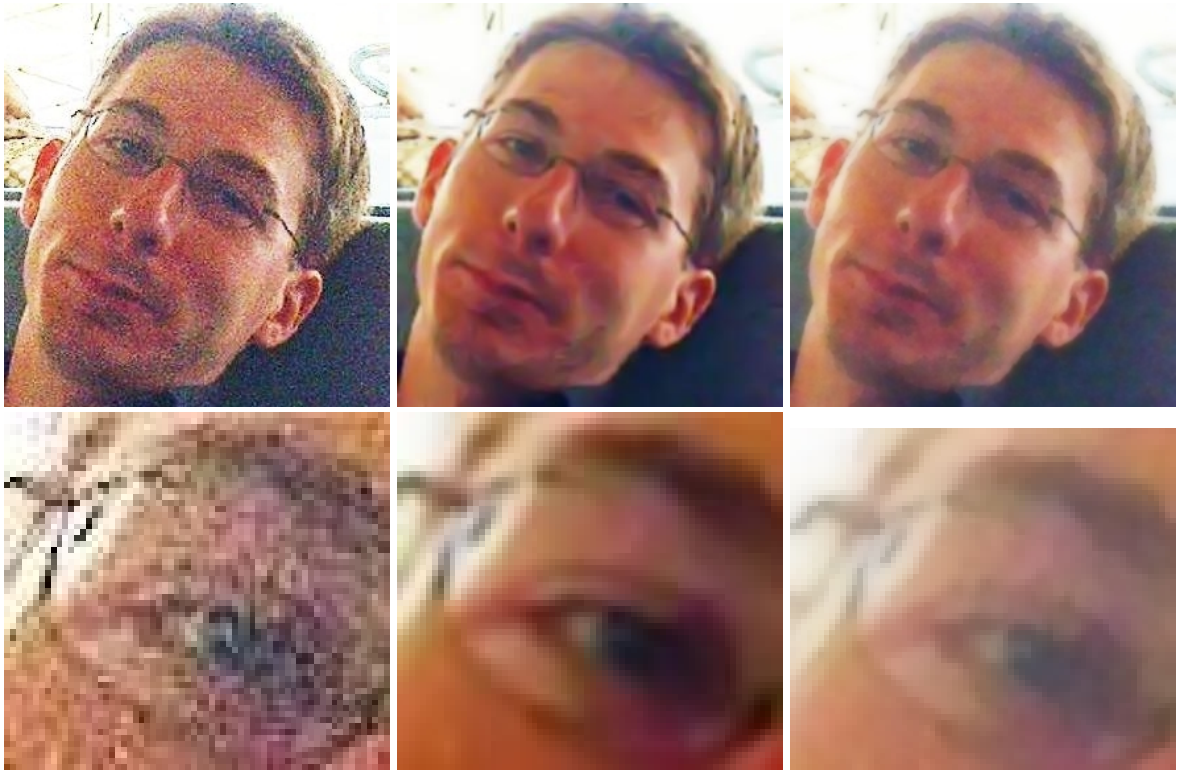


Figure 6: An image with real camera noise and filtering results. On the left one can see the original noisy image. In the middle the result by [20] and on the right we present our result. The result in the center is better, but about a minute was needed to compute it due to the optimization of a second order PDE, whereas our result needed less than a second of computation time. The lower row shows closeups of the upper one

5 Conclusion

The paper shows how to interpret image processing problems in terms of classical variational calculus. We have defined a class of functionals that, firstly, is able to incorporate different kinds of external information through two (or more) functions, which do not vary within a frame, with very high speed and, secondly, allows efficient computation by making use of the positiveness of the functionals. We have proved the mathematical correctness of the suggested methods. The algorithms have been presented precisely, which allows for easy reimplementations.

We have used the class of functionals to tackle two real world problems, for which we also show the results. We have shown that by making use of an external edge detector images can be smoothed while preserving edges using our method as well as an example of area-based (as compared to pixel-based) segmentation by incorporating a simple heuristic. The segmentation results are promising even though we only used color, edge, and background information. Incorporating other cues, e.g., motion detection or model knowledge will probably lead to even better results and is subject to ongoing research. These cues can easily be introduced into our functional by making use of the functions $k(\vec{x})$ and $p(\vec{x})$ in the same way that we demonstrated for the incorporation of edge information.

This yields methods for edge-preserving smoothing and cue-based segmentation which are an order of magnitude faster than methods based on nonlinear PDEs but still yield results with sufficient quality for image understanding purposes.



Figure 7: Examples of segmentation. **a)** and **b)** show the original image, the segmentation result, and the object probability for successful attempts. **c)** shows background and image for a failed attempt. Segmented regions are marked in a different color for visualization.

Acknowledgments

Funding by the Deutsche Forschungsgemeinschaft (MA 697/5-1, WU 314/5-2) is gratefully acknowledged. We thank Bram Bolder and Carsten Prodöhl for posing for the videos.

References

- [1] C.A.Z. Barcelos, M. Boaventura, and E.C. Silva. A well-balanced flow equation for noise removal and edge detection. *IEEE Trans. Image Processing*, 12(7):751–763, 7 2003.
- [2] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, Cambridge MA, 1987.
- [3] J. Canny. A computational approach to edge detection. *IEEE Trans. PAMI*, 8:679–698, 1986.
- [4] R. Courant and D. Hilbert. *Methods of mathematical physics. Volume II. Partial differential equations*. New York, Chichester, Brisbane : John Wiley & Sons, 1962.
- [5] D. Geman and G. Reynolds. Constrained restoration and the recovery of discontinuities. *IEEE Trans. PAMI*, 14:367–383, March 1992.

- [6] G. Gilboa, N. Sochen, and Y.Y. Zeevi. Image enhancement and denoising by complex diffusion processes. *IEEE Trans. PAMI*, 26(8):1020–1036, 2004.
- [7] G. Hewer, C. Kenney, and B. Manjunath. Variational image segmentation using boundary functions. *IEEE Trans. Image Processing*, 7(9):1269–1282, 1998.
- [8] B.B. Kimia and K. Siddiqi. Geometric heat equation and nonlinear diffusion of shapes and images. *CVGIP Image Understanding*, 64(3):305–322, 1996.
- [9] S.G. Krein, N.D. Kopachevsky, and Ngo Zuy Kan. *Operatornye metody v lineinoy gidrodinamike: Evolutsionnye i spektralnye zadachi*. Moscow, "Nauka", 1989. In Russian.
- [10] M. Lysaker, S. Osher, and Xue-Cheng Tai. Noise removal using smoothed normals and surface fitting. *IEEE Trans. Image Processing*, 13(10):1345–1357, 2004.
- [11] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problem. *Comm. Pure Appl. Math.*, 42:577–684, 1989.
- [12] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. PAMI*, 12:629–639, 1990.
- [13] M. Reed and B. Simon. *Methods of Modern mathematical physics: Functional analysis*, volume 1. Academic Press New York, London, 1972.
- [14] Olivier Renaud, Jean-Luc Starck, and Fionn Murtagh. Wavelet-based combined signal filtering and prediction. *IEEE Trans. SMC, Part B*, 35(6):1241–1251, 2005.
- [15] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.
- [16] P. Saint-Marc and G. Medioni. Adaptive smoothing for feature extraction. *Proc. DARPA Image Understanding Workshop*, pages 1100–1113, 1988.
- [17] E P Simoncelli and E H Adelson. Noise removal via Bayesian wavelet coring. In *Third Int'l Conf on Image Proc*, volume I, pages 379–382, Lausanne, 1996. IEEE Sig Proc Society.
- [18] J. Stoer, R. Bulirsch, R. Bartels, W. Gautschi, and C. Witzgall. *Introduction to numeric analysis*. Springer, 1992.
- [19] A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-posed Problems*. W. H. Winston, Washington D.C, 1977.
- [20] David Tschumperlé and Rachid Deriche. Vector-valued image regularization with PDEs: A common framework for different applications. *IEEE Trans. PAMI*, 27(4):506–517, 2005.

A Proof of lemma 1

From Poincaré's inequality [4] we have:

$$\|u\|_{\mathbb{L}_2}^2 \leq \frac{\langle u, 1 \rangle_{\mathbb{L}_2}^2}{\int_{\Omega} k \, d\vec{x}} + \frac{k_1}{p_0} \mu(\Omega) \int_{\Omega} p(\nabla u)^2 \, d\vec{x} \leq \quad (34)$$

$$\frac{\|1\|_{\mathbb{L}_2}^2}{\int_{\Omega} k \, d\vec{x}} \|u\|_{\mathbb{L}_2}^2 + \frac{k_1}{p_0} \mu(\Omega) \int_{\Omega} p(\nabla u)^2 \, d\vec{x} \leq a \|u\|_{\mathbb{H}^1}^2 \quad (35)$$

with $a := \max \left\{ \frac{\|1\|_{\mathbb{L}_2}^2}{\int_{\Omega} k \, d\vec{x}}, \frac{k_1}{p_0} \mu(\Omega) \right\}$, and $\mu(\Omega) := \int_{\Omega} d\vec{x}$

B Proof of lemma 2

$$y(x) := \langle x, y \rangle_E \quad (36)$$

This functional is bounded since

$$|y(x)| = |\langle x, y \rangle_E| \leq \|x\|_E \|y\|_E \leq a \|x\|_F \|y\|_E \quad (37)$$

From the Riesz representation theorem [13] we know that each bounded functional in a Hilbert space can be represented by a unique element of this space as

$$y(x) = \langle x, z \rangle_F \quad (38)$$

and from (36) we have (7)

C Proof of lemma 3

1. $\frac{1}{a} \|y\|_{\tilde{\mathbb{L}}_2} \leq \|Vy\|_{\tilde{\mathbb{H}}^1} \leq a \|y\|_{\tilde{\mathbb{L}}_2};$
2. $\langle y, Vy_1 \rangle_{\tilde{\mathbb{H}}^1} = \langle y, y_1 \rangle_{\tilde{\mathbb{L}}_2} = \langle y_1, y \rangle_{\tilde{\mathbb{L}}_2} = \langle y_1, Vy \rangle_{\tilde{\mathbb{H}}^1} = \langle Vy, y_1 \rangle_{\tilde{\mathbb{H}}^1};$
3. $\langle Vy, y \rangle_{\tilde{\mathbb{H}}^1} = \|y\|_{\tilde{\mathbb{L}}_2}^2 \geq 0.$

D Proof of lemma 4

1. $\forall x \in D(A) \quad \langle Ax, x \rangle_{\tilde{\mathbb{L}}_2} = \|x\|_{\tilde{\mathbb{H}}^1}^2 \geq \frac{1}{a} \|x\|_{\tilde{\mathbb{L}}_2}^2$
2. from Green's formula we have:

$$\langle f, \delta \rangle_{\tilde{\mathbb{L}}_2} = \langle Au, \delta \rangle_{\tilde{\mathbb{L}}_2} = \langle u, \delta \rangle_{\tilde{\mathbb{H}}^1} = \int_{\Omega} ku \delta \, d\vec{x} + \int_{\Omega} p \nabla u \cdot \nabla \delta \, d\vec{x} = \quad (39)$$

$$\int_{\Omega} (ku - \operatorname{div}(p \nabla u)) \delta \, d\vec{x} + \int_{\partial\Omega} p \frac{\partial u}{\partial n} \delta \, d\vec{x}. \quad (40)$$

Note that δ is an arbitrary smooth function and in case $p > 0$ we get (11) and (12).

E Proof of lemma 5

$$\|u\|_0^2 \leq \|u\|_s^2 \leq \gamma \|u\|_0^2 \quad (41)$$

with

$$\gamma := \max \left\{ 1, \frac{\sum_{i=2}^n \sup k_i(\vec{x})}{\inf(k_1)} \right\} \quad (42)$$