SIGNAL & IMAGE DENOISING

USING CONSTRAINED OPTIMIZATION

Michael Bronstein
Alexander Bronstein

SUPERVISED BY

Prof. Yehoshua Y. Zeevi

CONCLUDING REPORT FOR THE SEMINAR
ADVANCED TOPICS IN COMPUTER VISION

April - July 2001

I BACKGROUND

A. Adaptive diffusion equation

Denoising of images is an important task in image processing and analysis, and it plays a significant role in modern applications in different fields, including medical imaging and preprocessing for computer vision.

Consider an image contaminated by noise (without loss of generality, we assume additive noise):

$$I(\boldsymbol{x}) = f(\boldsymbol{x}) + \xi(\boldsymbol{x})$$

Our goal is to remove that noise, resulting in minimal damage to the image. Since in most cases the result consumer is human, the criterion for denoising fidelity would be the human visual perception of the result, rather than any of known mathematical criteria, such as minimal mean error (MSE) or minimal maximal difference (minimax).

The most naïve approach would be performing some kind of low-pass filtering on the image, e.g. by convolution of Gaussian. Perona and Malik [1] claim that low-pass filtering by Gaussian kernel convolution can be equivalently formulated as forward diffusion

$$\partial_t I(\boldsymbol{x};t) = c\nabla^2 I(\boldsymbol{x};t) \qquad c = \text{const}$$

where t is the time parameter proportional to the Gaussian standard deviation and c is the diffusion coefficient.

The disadvantage of low-pass filtering or homogenous adaptive diffusion in image enhancement applications is the fact that Gaussian blurring does not respect the natural edges of the image. Hence, it is impossible to use simple Gaussian blurring in order to enhance the natural features of the image, which the eye perceives the best.

ORIGINAL IMAGE **DESIRED RESULT** **GAUSSIAN BLUR RESULT**

Figure 1.1: Gaussian blurring result: both the edges and the background are blurred (this image is a visualization of the "tree" example given by Perona & Malik, created in PhotoShop).

The desired denoising algorithm must be edge-preserving, thus performing diffusion only in the regions divided by edges and leaving the edges untouched, what is termed as "strong intra-region diffusion" and "weak inter-region diffusion". The formulation of edge criterion is part of the solution rather than of the problem itself, since different edge indicators may be used, leading to different results.

One of the ways to implement such algorithm according to Perona and Malik [1] is by solving a non-homogenous diffusion equation.

$$\partial_t I(\boldsymbol{x};t) = c\nabla^2 I(\boldsymbol{x};t) \qquad c = \varphi\left(\|\nabla I\|\right)$$

The diffusion coefficient achieves high value (strong blurring) in the interior of each region and low value (weak blurring) on the edges. The edges are determined by local gradient value, so that the diffusion coefficient is a function of the gradient norm of the form:

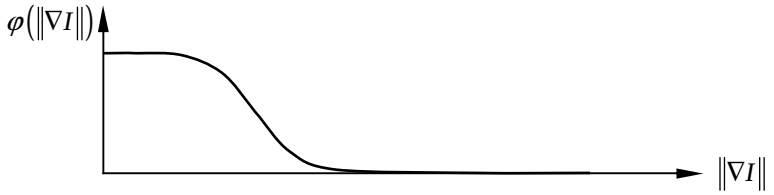


Figure 1.2: diffusion coefficient for adaptive PM diffusion.

The gradient norm is one of the most popular edge indicators in image processing, but it appears sometimes even inapplicable in case of high-amplitude noise presence.

In order to enhance the edges, an inverse process may be used. Inverse diffusion in general is obtained by running the diffusion process backwards in time, or equivalently, taking a negative constant in the homogenous diffusion equation

$$\partial_t I(\boldsymbol{x};t) = -c\nabla^2 I(\boldsymbol{x};t) \qquad c > 0$$

and is equivalent to the problem of Gaussian kernel deconvolution. Such problems are known as ill-posed for they are numerically unstable.

Consider some distortion process performed by convolving kernel h with the image, which gives by the convolution theorem,

$$g(\boldsymbol{x}) = i(\boldsymbol{x}) * h(\boldsymbol{x}) \quad \overset{\mathcal{F}}{\leftrightarrow} \quad G(\boldsymbol{\omega}) = I(\boldsymbol{\omega})H(\boldsymbol{\omega})$$

Our goal is to find the inverse operator h^{-1}, so that

$$i(\boldsymbol{x}) = g(\boldsymbol{x}) * h^{-1}(\boldsymbol{x}) \quad \overset{\mathcal{F}}{\leftrightarrow} \quad I(\boldsymbol{\omega}) = \frac{I(\boldsymbol{\omega})H(\boldsymbol{\omega})}{H(\boldsymbol{\omega})} = \frac{G(\boldsymbol{\omega})}{H(\boldsymbol{\omega})}$$

In case of Gaussian kernel, H is a low-pass filter, and its inverse kernel H^{-1} is potentially unstable.

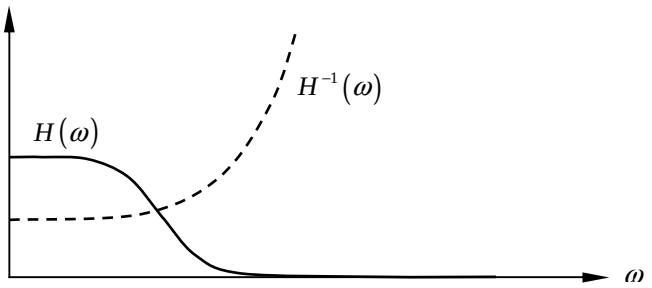
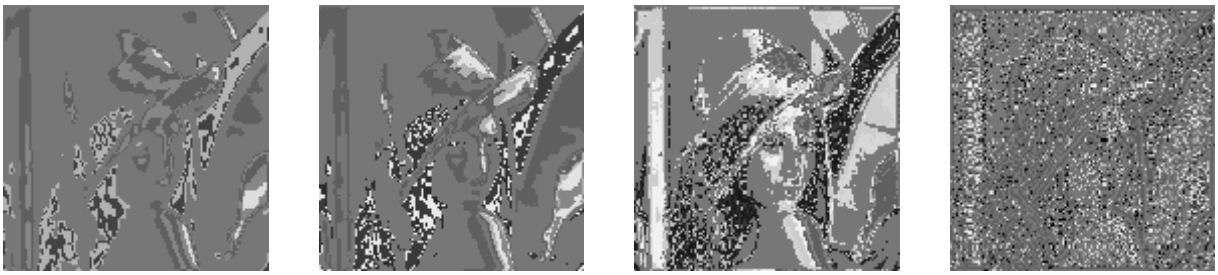


Figure 1.3: schematic plot of the 1D Gaussian kernel and its inverse in the frequency domain.

Similarly to deconvolution, inverse diffusion is an ill-posed problem and thus gives rise to numerical instability, demonstrated in **Figure 1.4.**:



ORIGINAL IMAGE **5 ITERATIONS** **10 ITERATIONS** **18 ITERATIONS**

Figure 1.4: example of inverse diffusion instability: application of inverse diffusion process on a blurry image of Lena results in certain improvement (after 10 iterations), but due to instability the image is completely distorted after few iterations (MATLAB simulation).

Generalized forward-and-backward diffusion proposed by Gilboa, Zeevi and Sochen [2,3] allows local inverse diffusion process to enhance the edges without destroying the stability of the diffusion equation. The diffusion coefficient in this process is negative at high gradient norm values:

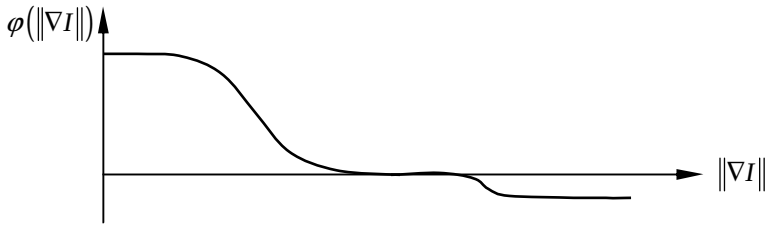
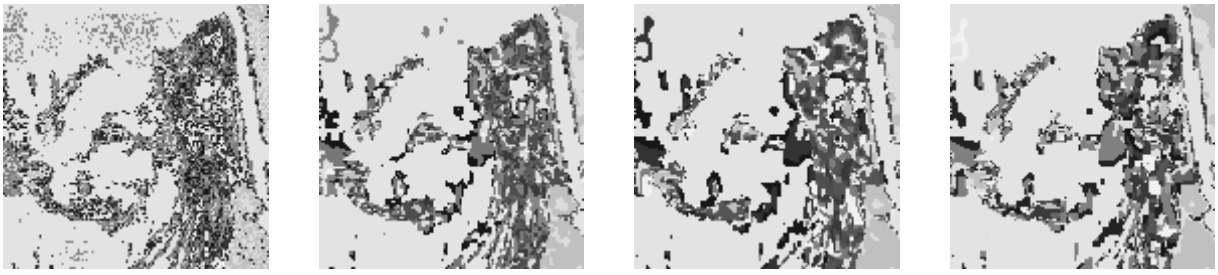


Figure 1.5: diffusion coefficient for generalized forward-and-backward diffusion.

As the result, generalized forward-and-backward diffusion algorithm is capable of not only removing intra-regional noise, but also of enhancing inter-region contrast by locally sharpening the edges.



ORIGINAL IMAGE **10 ITERATIONS** **20 ITERATIONS** **30 ITERATIONS**

Figure 1.6: application of generalized forward-and-backward diffusion for edge-preserving denoising (MATLAB simulation).

A general property of non-linear adaptive diffusion processes is that they encourage intra-region blurring and prevent inter-region blurring, thus preserving edges whilst enjoying stability guaranteed by the maximum principle. According to the maximum/minimum principle, no new local minima or maxima are introduced, in order not to produce artifacts in the diffused image. Moreover, the values of the global minimum and maximum along the evolution of the image are bounded by those of the initial state. These properties have crucial significance in medical image processing.

B. Representing images as embedded maps

Kimmel, Sochen and Malladi [4,5] propose a geometric approach, which assumes treating images as Riemannian manifolds embedded in multidimensional spaces. Such approach allows defining a generalization of Perona-Malik diffusion and other similar methods.

The key idea is the representation of an image as a manifold Σ, embedded in some space M. Since the coordinates of both manifold and the embedding space are curvilinear and not orthogonal in general, the metrics are not necessarily Euclidean, and are defined in most general case by the metric tensors $g_{\mu\nu} : \mu,\nu = 1,...\dim\Sigma$ and $h_{ij} : i,j = 1,...\dim M$. The distance between two points $p = \left(\sigma^1,\sigma^2\right)$ and $p + \left(d\sigma^1, d\sigma^2\right)$ on the manifold is given by $ds^2 = g_{\mu\nu} d\sigma^\mu d\sigma^\nu$. The inverse metric tensor is denoted by $g^{\mu\nu}$ and obeys $g^{\mu\nu} g_{\nu\gamma} = \delta^\mu_\gamma$.

The embedding is defined by a map $\mathbf{X} : \Sigma \to M$, which transforms the local coordinates into the coordinates of the embedding space. If the map $\mathbf{X}$ and the metric h are given, the metric g can be found by the pullback procedure:

$$g_{\mu\nu} = h_{ij}\left(\mathbf{X}\right)\partial_\mu X^i \partial_\nu X^i$$

This approach is a general framework for non-linear diffusion, which incorporates many methods such as the Perona and Malik algorithm, and also allows to derive new ones. Diffusion equations are derived by a minimization problem of the Polyakov functional, which depends on both manifold and the embedding space:

$$S\left[X^i, g_{\mu\nu}, h_{ij}\right] = \int d^m\sigma\sqrt{g}\, g^{\mu\nu}\partial_\mu X^i \partial_\nu X^j h_{ij}\left(\mathbf{X}\right)$$

resulting in the gradient-flow evolution equation given as a variational equation of the action functional S:

$$\partial_t X^i = -\frac{1}{2\sqrt{g}} h^{il}\frac{\delta S}{\delta X^l}$$

Particularly important case is the embedding of Σ in $\mathbb{R}^3$. Let σ^1,σ^2 be the local coordinates of the manifold and X^1, X^2, X^3 denote the global coordinates of the embedding space. The map $\mathbf{X} : \Sigma \to \mathbb{R}^3$ is given explicitly by

$$\left(X^1\left(\sigma^1,\sigma^2\right), X^2\left(\sigma^1,\sigma^2\right), X^3\left(\sigma^1,\sigma^2\right)\right)$$

We will discuss the simplest case, in which a grayscale image is represented as a graph in $\mathbb{R}^3$. Given the intensity level of the represented image as a function of the local coordinates, $I\left(\sigma^1,\sigma^2\right)$, the graph is defined by the mapping

$$\mathbf{X} : \left(\sigma^1,\sigma^2\right) \to \left(\sigma^1,\sigma^2, I\left(\sigma^1,\sigma^2\right)\right)$$

or, in more convenient notation as $\mathbf{X} = \left(x, y, I\right)$ replacing local coordinates with $\sigma^1 \equiv x$ and $\sigma^2 \equiv y$, respectively, as shown in **Figure 1.7.**

In this case the metric tensor is simply

$$g_{\mu\nu} = \begin{pmatrix} 1+I_x^2 & I_x I_y \\ I_x I_y & 1+I_y^2 \end{pmatrix}$$

and

$$ds^2 = \begin{pmatrix} dx & dy \end{pmatrix}\begin{pmatrix} 1+I_x^2 & I_x I_y \\ I_x I_y & 1+I_y^2 \end{pmatrix}\begin{pmatrix} dx \\ dy \end{pmatrix} = \left(1+I_x^2\right)dx^2 + 2I_x I_y dxdy + \left(1+I_y^2\right)dy^2$$

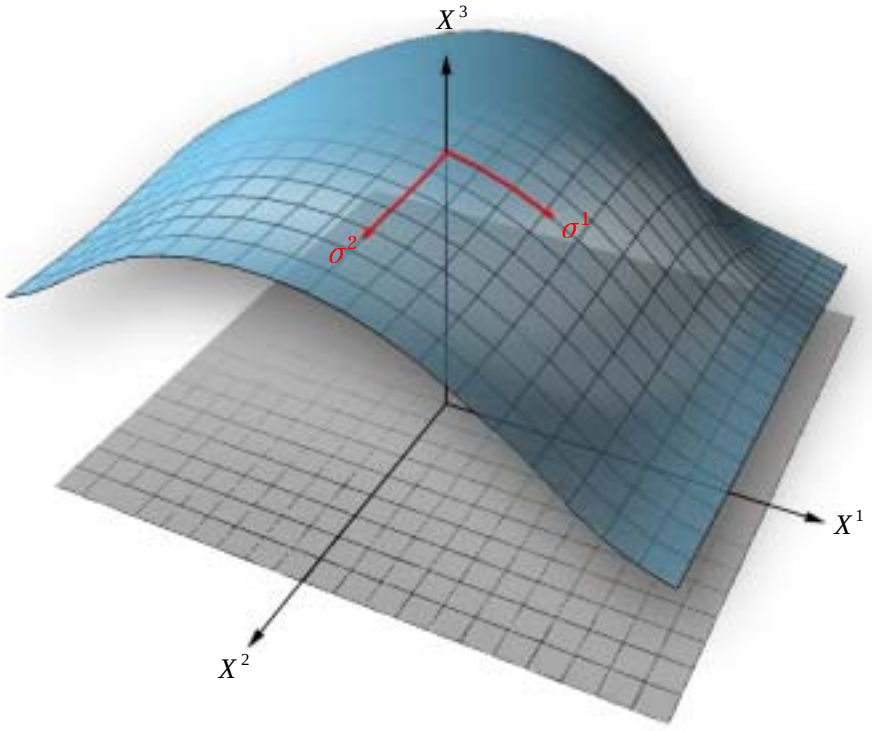


Figure 1.7: grayscale image represented as Riemannian manifold in 3D Euclidean space.

In case of an Euclidean embedding space, as in the discussed example, the flow minimizes functional S with respect to I and g, a particular case of the functional minimization shown above. The flow equation obtains the form

$$I_t = \Delta_g I \equiv \frac{1}{\sqrt{g}} \partial_\mu \left(\sqrt{g} g^{\mu\nu} \partial_\nu I \right)$$

where Δ_g denotes Beltrami operator, which is a generalization of Laplacian from planar surfaces to manifolds. In this way, Beltrami flow is a generalization of the non-linear diffusion proposed by Perona and Malik to the new representation scheme. The determinant g may be considered a generalized edge indicator.

Assuming representation of a grayscale image as a graph in $\mathbb{R}^3$, Beltrami flow can be written explicitly as

$$I_t = \Delta_g I = \frac{\left(1+I_y^2\right) I_{xx} - 2 I_x I_y I_{xy} + \left(1+I_x^2\right) I_{yy}}{\left(1+I_x^2+I_y^2\right)^2}$$

with initial condition $I(x,y;0) = I(x,y)$. The denominator is an edge indicator, similar to the gradient norm used by Perona and Malik and it dictates the flow strength along and across the edges.

C. Minimal surface flows

A result from differential geometry used by Kimmel et al is the fact that a graph has minimal surface if its mean curvature H equals zero. In the discussed case, the graph is $\left(x, y, I(x,y)\right)$, and its minimal curvature is given by

$$H = \frac{\left(1+I_y^2\right)I_{xx} - 2I_xI_yI_{xy} + \left(1+I_x^2\right)I_{yy}}{\left(1+I_x^2+I_y^2\right)^{3/2}}$$

This result leads to a flow towards minimal surface, usually termed as minimal curvature or minimal surface, given by the following evolution equation:

$$I_t = g\Delta_g I = \frac{\left(1+I_y^2\right)I_{xx} - 2I_xI_yI_{xy} + \left(1+I_x^2\right)I_{yy}}{1+I_x^2+I_y^2}$$

Both ideas of geometric representation and surface area minimization lead to a natural formulation of a constrained optimization problem, which we show in the next part.

II CONSTRAINED CURVE LENGTH MINIMIZATION - 1D

A. Optimization approach

In Part I we discussed adaptive diffusion methods and showed an approach derived from differential geometry, which in fact incorporates the majority of non-linear diffusion equations. In this part we propose to treat a particular application of adaptive diffusion for denoising from the point of view of optimization. We adopt the idea of representing images as surfaces in $\mathbb{R}^3$, which is a simplification of the approach proposed by Kimmel et al, but yet allows to give the problem a simple interpretation as an optimization problem. We first discuss a 1D model, and then generalize it in Part III for images.

Consider a step-function used as one-dimensional edge model, contaminated by additive Gaussian noise:

$$x(t) = f(t) + \xi(t)$$

(f denotes the step and ξ denotes the noise). Minimal surface flows discussed before minimize the surface area of the signal under appropriate constraints. In one-dimensional case, the equivalent of the surface area is the curve length.

Hence, optimally denoised signal should possess the minimal curve length. This result is quite intuitive since the curve length of a "clean" step is smaller than of a noisy one. However, if no constraints are posed, the optimal solution would be a constant function, i.e. the step would be completely blurred, which is obviously not desirable. To prevent such effect, edge-preserving constraints should be formulated.

B. Curve length in a 1D case

In this context, we discuss only discrete signals (simulated or obtained by sampling of continuous functions). Let $\boldsymbol{x} \in \mathbb{R}^n$ be a vector representation of signal $x(t) = f(t) + \xi(t)$ sampled at equal intervals ε. The curve length of $\boldsymbol{x}$ is given by

$$C(\boldsymbol{x}) = \sum_{i=1}^{n} \sqrt{(x_{i+1} - x_i)^2 + \varepsilon^2}$$

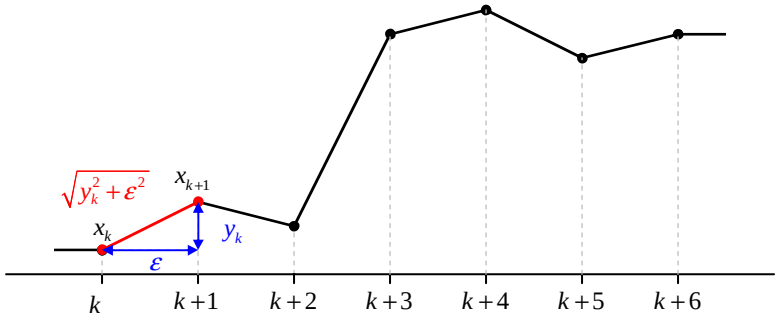


Figure 2.1: calculation of the curve length of a discrete function

Defining the vector of differences $y_i = x_{i+1} - x_i$, we can express the curve length using convex approximation of the absolute value function

$$C(\boldsymbol{y}) = \sum_{i=1}^{n} \sqrt{y_i^2 + \varepsilon^2} \overset{\varepsilon \to 0}{\approx} \|\boldsymbol{y}\|_1$$

The difference vector $\boldsymbol{y} \in \mathbb{R}^{n-1}$ can be expressed in matrix notation defining a circulant Toeplitz matrix $A \in \mathbb{R}^{n-1 \times n}$ of the form

$$A = \begin{pmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & 0 & -1 & 1 \end{pmatrix}$$

so that $\boldsymbol{y} = A\boldsymbol{x}$.

We shall note that such formulation is merely for notation convenience but may complicate the implementation in case of large signals due to large size of the matrix A. Though this aspect is out of the scope, a straightforward calculation of $\boldsymbol{y}$ may be used instead, significantly decreasing time and space complexity of the calculation.

C. Constrained optimization problem formulation

Similar to adaptive diffusion algorithms, we wish a small change of inter-region values (across the edge) and large change of the inter-region values on two subsequent iterations. This effect can be achieved by adding a penalty factor to the cost function.

Let $\boldsymbol{x}^k$ be the vector on current iteration and $\left\{\boldsymbol{x}^i\right\}_{i=k-N-1}^{k-1}$ be the vectors on some previous N iterations. The modification of $\boldsymbol{x}^k$ w.r.t. $\boldsymbol{x}^{k\text{-}N-1}$ should be proportional to some weight determined by an edge criterion, such as the gradient norm. Thus, a penalty factor is

$$\sum_{i=1}^{n} w_i \left(x_i^{k\text{-}N\text{-}1} - x_i^k \right)^2$$

The whole cost function consists is of the form:

$$F(\boldsymbol{x}) = \sum_{i=1}^{n} \sqrt{(A\boldsymbol{x})_i^2 + \varepsilon^2} + \lambda \sum_{i=1}^{n} w_i \left(x_i^{k\text{-}1} - x_i^k \right)^2$$

where the first factor is the curve length and the second factor is the penalty multiplied by a regulator λ.

The weight vector $\boldsymbol{w}$ is calculated according to the initial vector value $\boldsymbol{x}^0$ and is a function of the squared gradient vector (discrete derivative in a 1D case) of the form

$$w \equiv \varphi(\nabla \boldsymbol{x}) = e^{\beta^{-1}(\nabla x)^2}$$

(the notation implies coordinate-wise squared $\nabla \boldsymbol{x}$ and coordinate-wise application of the function). The parameter β is selected empirically.

Thus, very large penalty is given for large changes across the edge (where $\boldsymbol{w}$ is large) and smaller penalty is given for large intra-region changes (where $\boldsymbol{w}$ is small).

The minimization of the cost function $F(\boldsymbol{x})$ is performed by a quasi-Newtonian or other efficient optimization method. We have adopted the BFGS algorithm for simulations.

D. Step denoising simulation

The algorithm was tested on a 1D edge, modeled by hard- and soft-threshold functions. The steps were contaminated by additive Gaussian noise. Empiric parameter β was selected to obtain a weight higher by several orders across the edge than near the edge.

1000 iterations of the curve length minimization were performed using the BFGS algorithm.

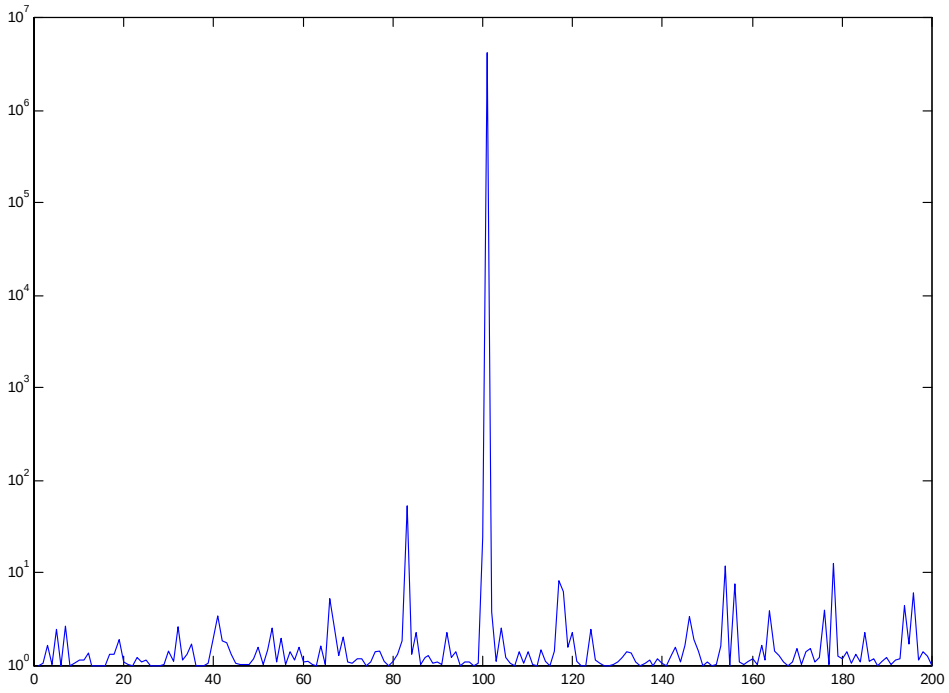


Figure 2.2: constraint weight plotted on logarithmic scale (obtained for sigmoidal step, β=0.015)

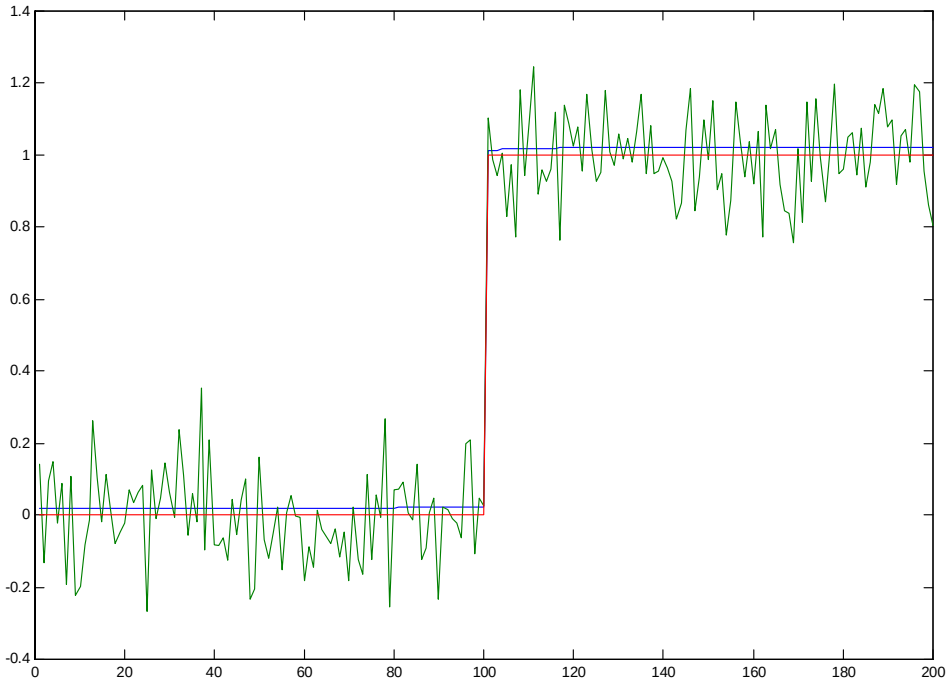


Figure 2.3: Ideal step (dotted red) contaminated by additive Gaussian noise (green) and restored by constrained curve length minimization (blue).

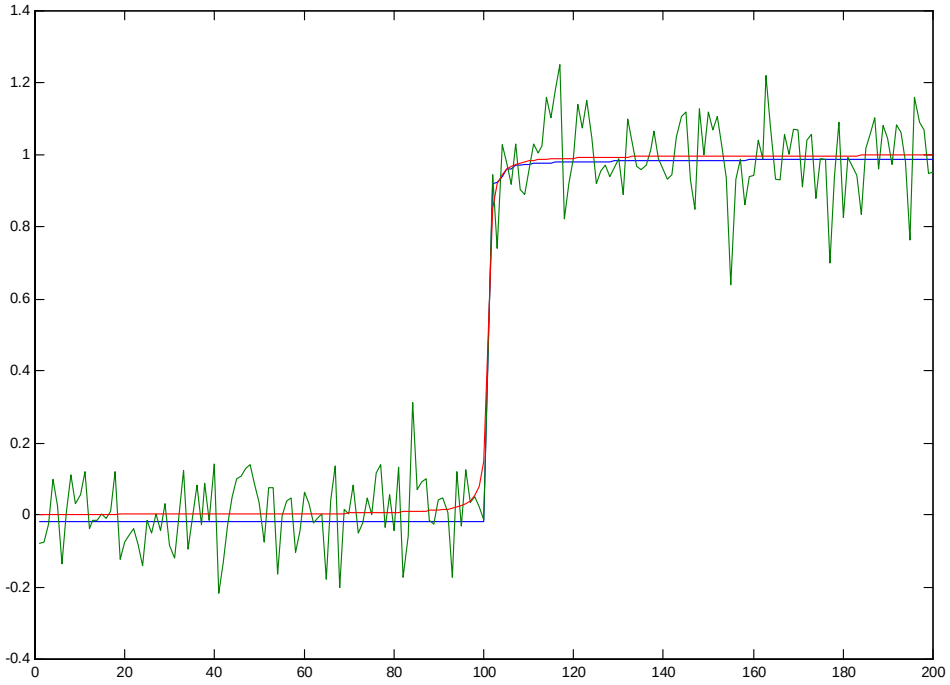


Figure 2.4: Sigmoidal step (dotted red) contaminated by additive Gaussian noise (green) and restored by constrained curve length minimization (blue).

III CONSTRAINED SURFACE AREA MINIMIZATION - 2D

A. Generalization of the curve length minimization

The most natural generalization of the demonstrated approach would be its application for 2D signals (images). Minimization of the curve length in this case is replaced by surface area minimization, similar to the procedure performed by minimal surface flows.

We should note that the most common way of approximating the area of a discrete 2D signal by dividing it into triangles is too complicated and thus is not desirable. Instead, other criterions that measure the "irregularity" of the images can be used. One such criterion is shown in (B). The advantage of such criterion is the fact that though it does not give the best approximation of the surface area, it leads to simple analytic expression and fast optimization.

B. Surface area of a 2D signal

A previously, we discuss only discrete signals (simulated or obtained by sampling of continuous functions). Without loss of generality, let $X \in \mathbb{R}^{n\times n}$ be a square matrix representing a 2D signal $\boldsymbol{x}(\boldsymbol{a}) = \boldsymbol{f}(\boldsymbol{a}) + \boldsymbol{\xi}(\boldsymbol{a})$ sampled at equal horizontal and vertical intervals ε. The surface area of X may be approximated by

$$S(X) = N^2\varepsilon^2 + \varepsilon\sum_{i,j=1}^{n-1}\left(\left|x_{i+1,j} - x_{i,j}\right| + \left|x_{i,j+1} - x_{i,j}\right|\right)$$

This is certainly not the best way to approximate the surface area, but it serves our needs ($S(X)$ is large for noisy images and small for clean ones) and as will be shown later, leads to simple analytic expressions in the optimization problem formulation.

A calculation example is shown in **Figure 3.1.**:

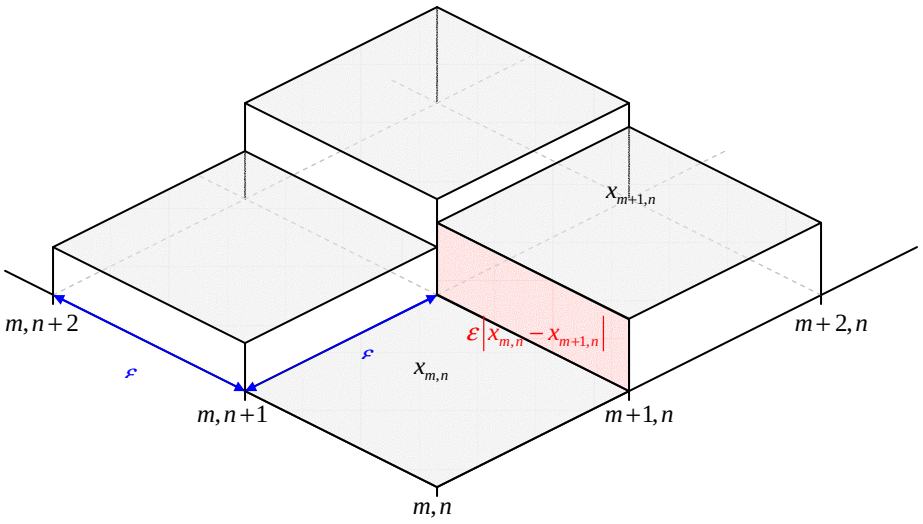


Figure 3.1: calculation of the surface area of a discrete 2D signal

Using column and row stack notation, horizontal and vertical difference matrices may be expressed in a simple form. Let $C \in \mathbb{R}^{n-1 \times n}$ be a circulant Toeplitz matrix

$$C = \begin{pmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & 0 & -1 & 1 \end{pmatrix}$$

Using this matrix, horizontal and vertical difference operators are given by $A = I_{n \times n} \otimes C$ and $B = C \otimes I_{n \times n}$, respectively, in row-stack notation ($\otimes$ denotes Krönecker tensor product). Applying these operators to parsed image matrix (row-stack) we obtain the vectors $\boldsymbol{y}_i^1 = (A\boldsymbol{x})_i$ and $\boldsymbol{y}_i^2 = (B\boldsymbol{x})_i$.

We should note that in case of large images stack representation is problematic due to large memory consumption, however it is very efficient in case of small images that were tested in the scope of this work. To reduce the matrix operator size, it is possible to use the separability of Krönecker product and apply separable operator on matrices.

C. Constrained optimization problem formulation

Similar to the one-dimensional case, we define the cost function as a sum of the surface area and the penalty factor, which expresses the constraint and prevents edge blurring. Using the expressions derived in (B), the optimization is performed on a parsed image in row-stack representation.

Let $\boldsymbol{x}^k$ be the row-stack vector on current iteration and $\{\boldsymbol{x}^i\}_{i=k-N-1}^{k-1}$ be the vectors on previous N iterations. The modification of $\boldsymbol{x}^k$ w.r.t. $\boldsymbol{x}^{k-1}$ is proportional to some weight

$$\sum_{i=1}^{n^2} w_i \left(x_i^{k-1} - x_i^k \right)^2$$

The whole cost function is of the form:

$$F(\boldsymbol{x}) = N^2\varepsilon^2 + \varepsilon \sum_{i=1}^{(n-1)\times n} \left(\left| (A\boldsymbol{x})_i \right| + \left| (B\boldsymbol{x})_i \right| \right) + \lambda \sum_{i=1}^{n^2} w_i \left(x_i^{k-1} - x_i^k \right)^2$$

and in order to make the cost function convex, we use a convex approximation of the absolute value function:

$$F(\boldsymbol{x}) = N^2\varepsilon^2 + \varepsilon \sum_{i=1}^{(n-1)\times n} \left(\sqrt{\left| (A\boldsymbol{x})_i \right| + \Delta^2} + \sqrt{\left| (B\boldsymbol{x})_i \right| + \Delta^2} \right) + \lambda \sum_{i=1}^{n^2} w_i \left(x_i^{k-1} - x_i^k \right)^2$$

where the first factor is the convex approximation of the curve length ($\Delta^2 << 1$) and the second factor is the penalty multiplied by a regulator λ.

The weight matrix W is calculated according to the initial image value X^0 and is a function of the horizontal and vertical derivatives. In order to avoid overemphasis of corners, we used the weigh

$$w \equiv \varphi\left(\partial_x X, \partial_y X\right) = e^{\beta^{-1}(\partial_x X)^2} + e^{\beta^{-1}(\partial_y X)^2}$$

instead of the gradient norm. The parameter β is selected empirically. The matrix W is parsed into a row-stack vector $\boldsymbol{w}$.

D. 2D Step denoising simulation

Surface area minimization algorithm was tested on a 2D step contaminated by additive Gaussian noise. Empiric parameter β was selected to obtain a weight higher by several orders across the edge than near the edge.

1500 iterations of surface area minimization were performed using the BFGS algorithm.

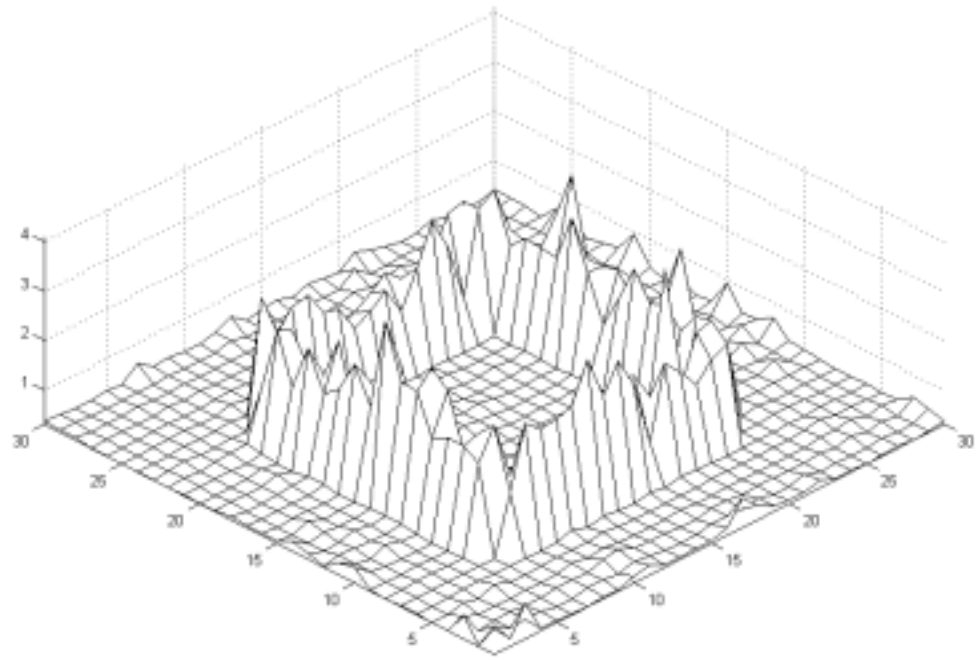


Figure 3.2: logarithm of the constraint weight matrix (β=0.05)

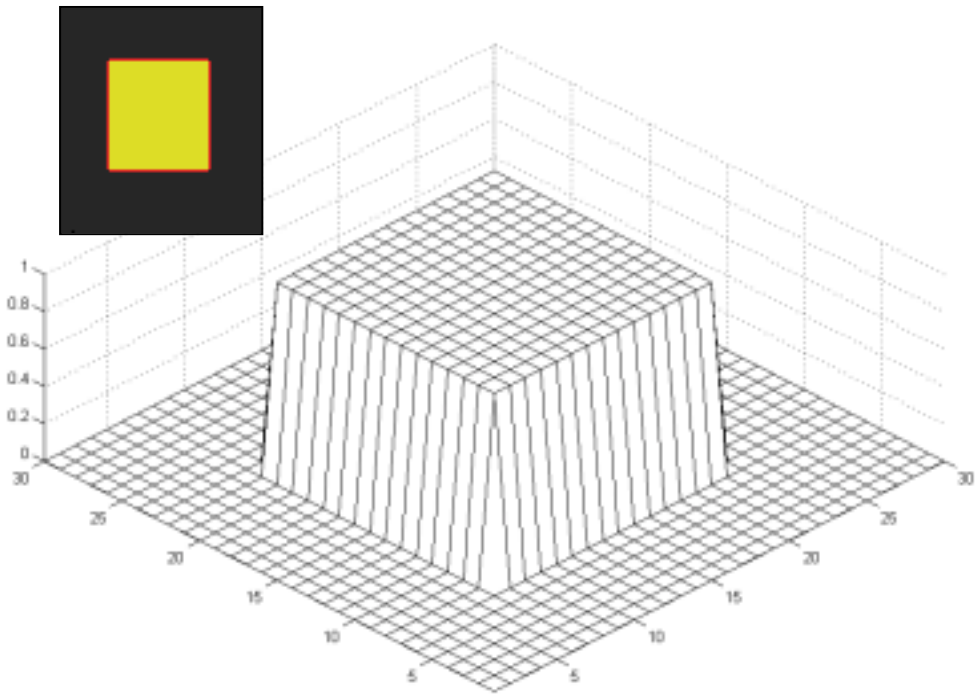


Figure 3.3: clean 2D step.

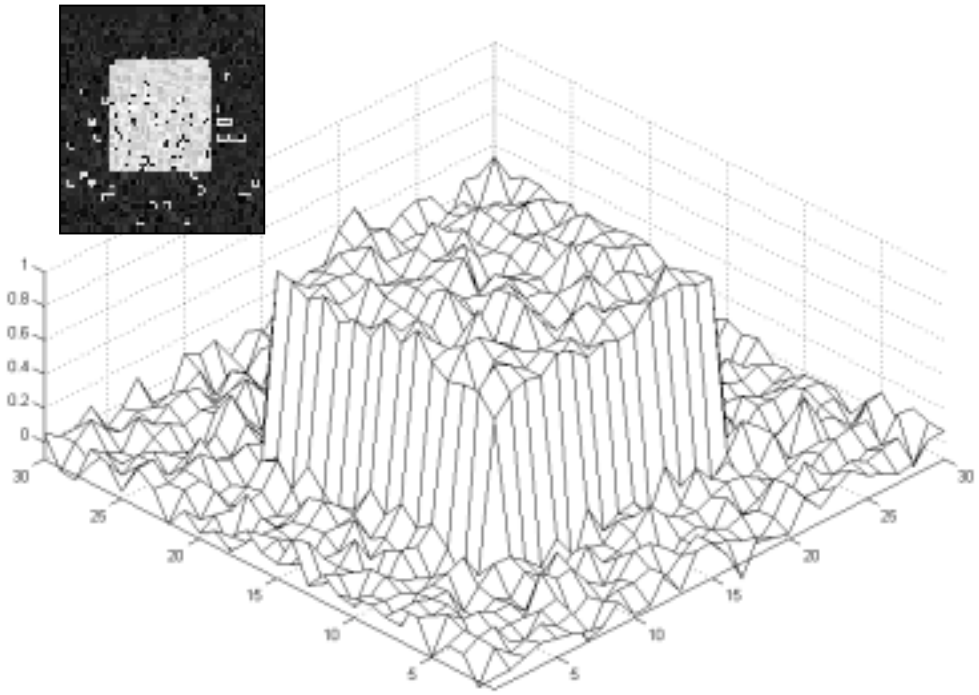


Figure 3.4: noisy step contaminated by Gaussian noise

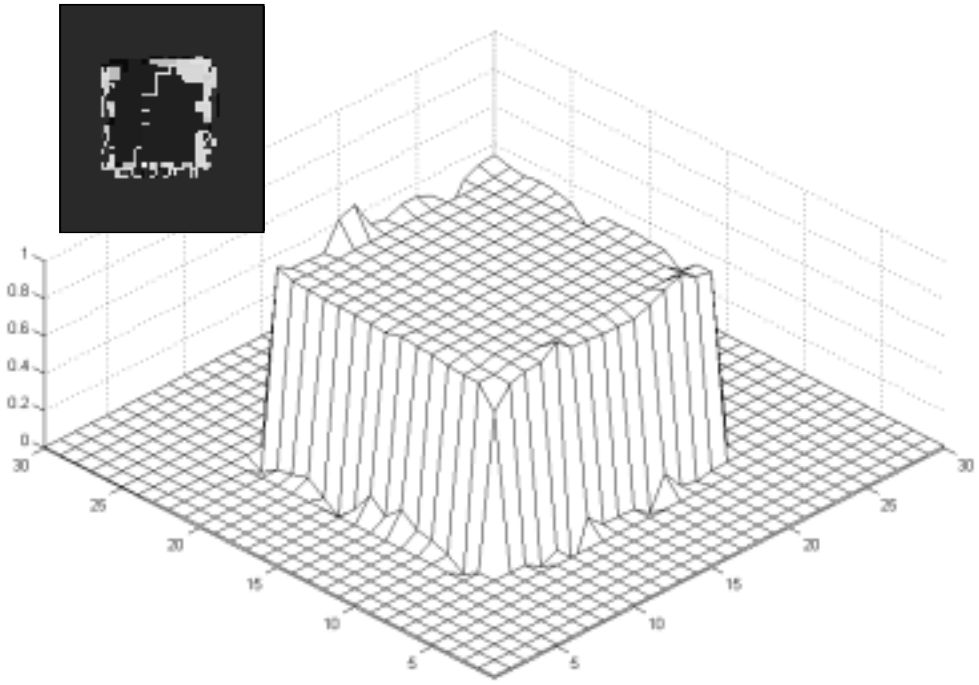


Figure 3.5: denoised step after 1500 quasi-Newtonian iterations.

RESULTS & CONCLUSIONS

IV

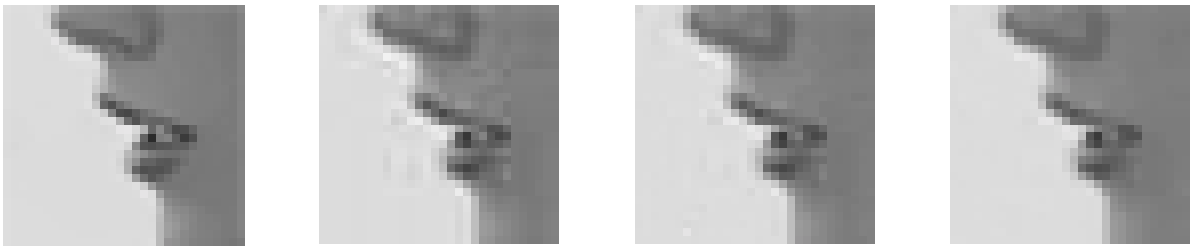
A. Removal of compression artifacts in lossy JPEGs

To demonstrate the ability of the proposed algorithm, we have performed several "real-life" denoising tests. One of specific applications of image denoising is the removal of artifacts caused by low-quality lossy JPEG compression. Kimmel et al [4,5] demonstrate an application of Beltrami flow for low-quality color JPEG image enhancement.

We find this application as probably the most suitable denoising case for edge-preserving adaptive diffusion algorithms, since despite the noise, the natural edges of the image are almost not damaged. This allows to use gradient-norm-based edge indicators, without being afraid of destroying natural edges or creating artificial ones due to misinterpretation of the noise as image edges.

We have taken an image of female face fragment similar to that used by Kimmel and artificially introduced artifacts by compressing it in JPEG format with the minimal possible compression quality using PhotoShop. The result was a noisy image, in which however the edges are quite clear despite the noise.

150 iterations of penalty aggregate function optimization were performed by quasi-Newtonian method.



CLEAN IMAGE **LOSSY IMAGE** **50 ITERATIONS** **150 ITERATIONS**

Figure 4.1: surface minimization progress.

Another test of image denoising as a way to compensate lossy JPEG compression artifacts was performed on an image of the character '*B*', shown in **Figure 4.2.** This image was contaminated by noise as the result of compression, however, as in previous example the edges are clearly perceptible.

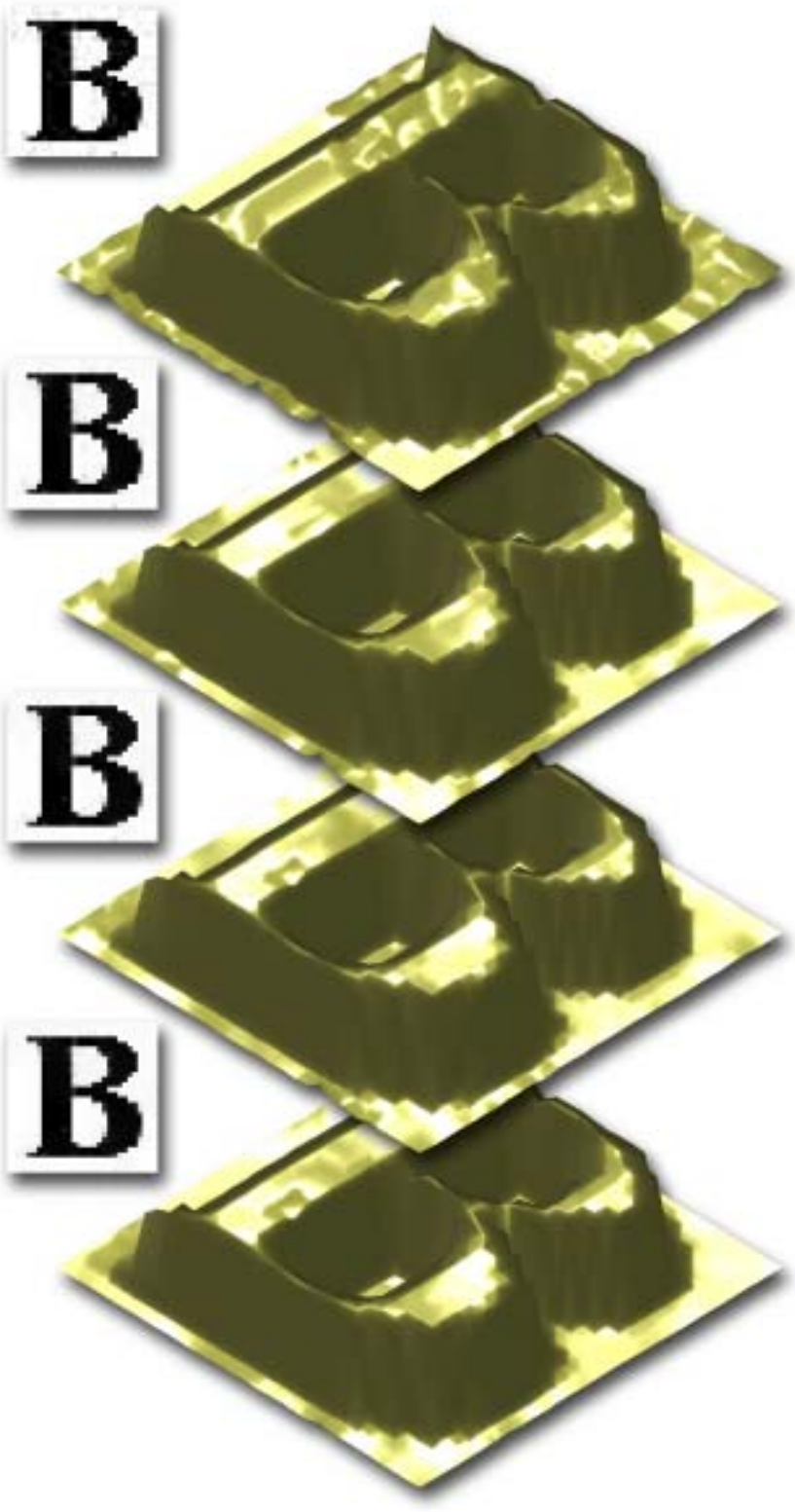


Figure 4.2: surface evolution: original image, rendered as surface (top) and surface area minimization result after 500 iterations (bottom).

B. Summary

It can be seen that the method of constrained surface optimization is equivalent to adaptive diffusion in the effect that it produces and hence can be used along with other methods of non-linear diffusion for various problems, including image denoising.

The dimension of the optimization problem equals the number of pixels in the image and thus we have a high-dimensional problem in case of large images. This may require efficient methods of high-dimensional constrained optimization.

One of the main problems of the described approach is the need for a *bona fide* edge indicator and the ability to formulate good constraints, which would not damage the edges on one hand, and would not prevent intra-region surface minimization on the other. This is a problem of edge-preserving algorithms in general, and particularly of those based on gradient norm as edge indicator, for in presence of high-amplitude noise artificial edges contributed by noise may be interpreted as actual edges of the image.

Since the optimization approach was derived from the geometric scheme as a very particular embedding case, other embeddings may be formulated as optimization problems as well, resulting in “optimizational” formulation of different flows.

REFERENCES

[1] P. Perona, J. Malik, *Scale-space and edge detection using anisotropic diffusion*, Transactions on Pattern Analysis and Machine Intelligence, vol. 12 No. 7 July 1990

[2] G. Gilboa, Y.Y. Zeevi, N. Sochen, *Signal and image enhancement by a generalized forward-and-backward adaptive diffusion process.*

[3] G. Gilboa, Y.Y. Zeevi, N. Sochen, *Anisotropic selective inverse diffusion for signal enhancement in the presence of noise*.

[4] R. Kimmel, N. Sochen, R. Malladi, *From high energy physics to low level vision.*

[5] R. Kimmel, R. Malladi, N. Sochen, *Images as embedded maps and minimal surfaces: movies, color, texture and volumetric medical images.*

[6] N. Sochen, G. Gilboa, Y.Y. Zeevi, *Color image enhancement by a forward-and-backward adaptive Beltrami flow.*

[7] G. Gilboa, Y.Y. Zeevi, N. Sochen, *Complex diffusion process for image filtering*.

[8] *Superresolution using adaptive diffusion*, Undergraduate project (supervised by G. Gilboa), VISL.

[9] G. Gilboa, *PDEs in image processing*, (presentation)

[10] G. Gilboa, Y.Y. Zeevi, *Superresolution by PDEs*, (presentation)

[11] C. T. J. Dodson, *Introducing surfaces*

[12] Guggenheimer, *Differential Geometry*

[13] M. Spivak, *Differential geometry*

[14] A. Goetz, *Introduction to differential geometry*