DIFFUSIONS AND CONFUSIONS
IN SIGNAL AND IMAGE PROCESSING

N. Sochen[1], R. Kimmel[2] and A. M. Bruckstein[2]
[1]*Department of Applied Mathematics*
School of Mathematical Sciences
Tel Aviv University
Tel Aviv 69978, Israel
email: sochen@math.tau.ac.il

[2]*Department of Computer Science*
Technion–Israel Institute of Technology
Technion City, Haifa 32000, Israel
email: ron,freddy@cs.technion.ac.il

Abstract. In this paper we link, through simple examples, between three basic approaches for signal and image denoising and segmentation: 1)PDE axiomatics 2) energy minimization 3) adaptive filtering. We show the relation between PDE's that are derived from a master energy functional, i.e. the Polyakov harmonic action, and non-linear filters of robust statistics. This relation gives a simple and intuitive way of understanding geometric differential filters like the Beltrami flow. The relation between PDE's and filters is mediated through the short time kernel.

1. Introduction

Averaging is a standard procedure for smoothing noisy data and summarizing information, but it can have rather dangerous and misleading results. Outliers, even if rare by definition, can distort the results considerably if they are given similar weights to "typical" data. Such concerns led to the development of so-called robust estimation procedures in statistical data analysis, procedures that data-adaptively determine the "influence" each data point will have on the results. Only recently were such ideas and methods imported to signal and image processing and analysis (Geman and Reynolds, 1992; Black, Sapiro, Marimont and Heeger, 1998; Comaniciu and Meer, 1999). The application of the robust statistics ideas in signal and image analysis lead to the introduction of various non-linear filters. To fix ideas and get a perspective on the serious problems that must be addressed we shall consider below a series of simple examples.

A seemingly different approach to denoising and segmentation is based on geometric properties of signals. The filtering is done, in this approach, by solving a non-linear Partial Differential Equation (PDE).

The derivation of the PDE is based either on axioms and requirements, such as invariance, separability (Perona and Malik, 1990; Alvarez et al., 1993; Mumford and Shah, 1985) etc., or as a by product of a minimization process for an energy functional (Mumford and Shah, 1985; Rudin Osher and Fatemi, 1992; Sochen, Kimmel and Malladi, 1998).

In this paper we discuss, through simple examples, the intimate connection between the above-mentioned signal and image processing methodologies: 1)PDE axiomatics 2) energy minimization 3) adaptive smoothing filters. We show the relation between PDE's that are derived from a master energy functional and non-linear filters. This relation gives a simple and intuitive way of understanding the Beltrami flow, and connects between geometric differential filters and classical linear and non-linear filters.

We show that the Beltrami flow, which results from the minimization process of the Polyakov action, is related to non-linear filters of special type upon choosing a L_γ induced metric and after discretization of the corresponding partial differential equation. A different non-linear filter is constructed via a short time analysis of the same PDE. The short time approach is different since we analyze and eventually discretize *the solution* of the differential equation. It is also more general since we treat a variety of flows by not specifying the explicit form of the metric at the outset. In this way we have a clear intuitive understanding of the adaptive averaging as a Gaussian weight function on the manifold that is defined by the data. The flows differ in the geometry attributed to the manifold through different choices of the metric.

The paper is organized in an increasing order of technicality. In Section 2 we discuss simple examples that make the basic idea clear. In Section 3 we deal with the more technical considerations of the short time analysis and show how different approaches are related to, or special cases of, this framework. We construct the appropriate non-linear filter for the challenging problem of the averaging of constrained features in Section 4. Image denoising is discussed and demonstrated in Section 5 and our conclusions are summarized in Section 5. Few of the more technical computations can be found in the Appendix.

2. Averaging Data for Smoothing and Clustering

To make our presentation as simple as possible we first limit our discussion to sets of scalar and vectors. One dimensional signals are analyzed next, and few remarks on higher dimension signals, e.g. images, can be found in the last subsection.

2.1. Averaging Scalar variables (values in $\mathbb{R}$)

Suppose we are given a set of real numbers $\{x_1, x_2, \ldots x_N\}$ and we would like to provide someone with a typical representative value that somehow describes these numbers. A natural choice would be their average, i.e.

$$\overline{x} = \sum_{i=1}^{N} \left(\frac{1}{N}\right) \cdot x_i \ . \tag{1}$$

While for sets like $\{2.83, 3.7, 3.22, 2.97, 3.05\}$ this is a reasonable choice, it clearly is not for a set of values comprising the number 10^{10} and a thousand values in the interval $(1-\epsilon, 1+\epsilon)$ for $\epsilon = 0.01$. In this second case, a typical number in the set is around 1 and 10^{10} may be regarded as an outlier to be either discarded or given "special consideration". To deal with such situations we could propose the following process of "smoothing" the initial set of values to produce new sets that more clearly exhibit the "inner structure" of the values in this set. For each $i = 1, 2, \ldots N$ do

$$x_i^{\text{new}} = (1-\alpha)x_i^{\text{old}} + \sum_{\substack{j=1 \\ i \neq j}}^{N} w_i(x_i^{\text{old}}, x_j^{\text{old}})x_j^{\text{old}}, \tag{2}$$

where $0 < \alpha < 1$. In matrix form

$$\mathbf{X}^{new} = \begin{pmatrix} 1-\alpha & w_1(x_1,x_2) & \cdots & w_1(x_1,x_N) \\ w_2(x_2,x_1) & 1-\alpha & \cdots & w_2(x_2,x_N) \\ \vdots & \vdots & \ddots & \vdots \\ w_N(x_N,x_1) & w_N(x_N,x_2) & \cdots & 1-\alpha \end{pmatrix} \mathbf{X}^{old}, \tag{3}$$

where $\mathbf{X}^T = (x_1, x_2, \cdots, x_N)$. The weights may be chosen so that

$$\sum_{\substack{j=1 \\ j \neq i}}^{N} w_i(x_i, x_j) = \alpha,$$

and $w_i(x_i, x_j)$ reflects how much influence x_j has on x_i based, say, on "how far" or "how different" x_j is from x_i. We further assume that the "distance" is always positive and symmetric. For example, we could choose

$$w_i(x_i, x_j) = \frac{\beta_i}{1 + |x_j - x_i|^\gamma} \qquad i \neq j,$$

with

$$\beta_i = \alpha \left(\sum_{\substack{j=1 \\ j \neq i}}^{N} \frac{1}{1 + |x_j - x_i|^\gamma} \right)^{-1} .$$

Such a choice will make nearby points more "influential" on the displacement of x_i toward its new location. If, however, we shall choose $w_i(x_i, x_j) = \frac{\alpha}{N-1}$ for all i, the dynamics of update becomes

$$x_i^{new} = (1-\alpha)x_i^{old} + \frac{\alpha}{N-1} \sum_{j \neq i} x_j^{old}, \tag{4}$$

and all points converge towards the mean of the initial location. Indeed we will have in this case

$$\mathbf{X}^{new} = \underbrace{\begin{pmatrix} 1-\alpha & \frac{\alpha}{N-1} & \cdots & \frac{\alpha}{N-1} \\ \frac{\alpha}{N-1} & 1-\alpha & & \frac{\alpha}{N-1} \\ \frac{\alpha}{N-1} & \cdots & & 1-\alpha \end{pmatrix}}_{\mathbf{B}} \mathbf{X}^{\mathbf{old}}, \tag{5}$$

and the circulant matrix here $\mathbf{B}$ is diagonalized by the Fourier Transform. It is easy to see that repeated application of the update rule (5) will yield asymptotically the vector

$$\frac{1}{N} \langle [1, 1, \ldots, 1], [x_1, x_2, \ldots, x_N] \rangle \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix},$$

where $\langle V, W \rangle$ is the scalar product of vectors.

Here the linear analysis was applicable while in the general case it is quite difficult to say what will be the exact dynamics of the set of points. However, if the set of points would comprise two subsets of values, one clustered around 1 and the other around 10^{10} then the corresponding non-linear dynamics will have the two clusters converging to the two centroids (averages) since points from the other set will be "deweighted" by about $\frac{1}{1+10^{10}}$. In fact this would precisely be the behavior if we choose the weight function $w(x, y)$ to be

$$(w(x, y) = (1 - \chi \{ d(x, y) > \text{“}Th''\}) \beta,$$

where $d(x, y)$ is the distance between x and y. Here χ is the indicator function for the predicate in the curly brackets and the threshold

"Th" can be chosen as 10,000. Therefore, we have seen that we can devise an averaging procedure via the "weight" or distance functions that will accomplish the following: If the data appears in well-defined (separated) clusters of points with distances between clusters of "Th" or more, the result of iterative application of the smoothing process will be a set of centroids that are typical values (in fact the averages) of each cluster. The same type of effect is achieved without the need of a threshold parameter with the weights $w_j(x_i, x_j) = \frac{1}{1+d(x_i,x_j)^\gamma} \cdot \beta$, for γ some positive constant. This is very nice since we have a data-adaptive smoothing process that provides an effective clustering of the data points into a variable size set of typical values. See Figure 1.

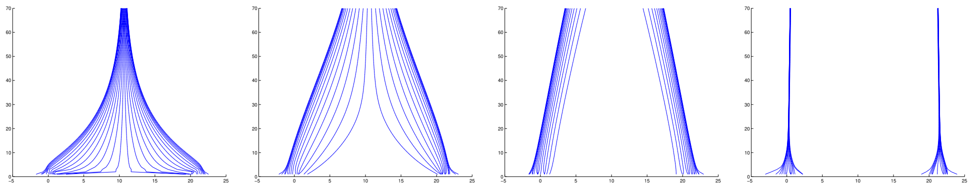

Figure 1. Random points on $\mathbb{R}^1$ propagate via adaptive averaging with $\gamma = 0, 1, 1.2, 2$ (left to right)

2.2. Averaging Vectors (values in $\mathbb{R}^d$)

Given N points in $\mathbb{R}^d$ we construct a smoothing process that moves each point towards the centroid of its cluster (see Fig. 2). Let $\mathbf{X}_i$ $i = 1, 2, \ldots, N$ be N vectors where $\mathbf{X}_i = (x_i^1, x_i^2, \cdots, x_i^d)^T$. We average each point in its local neighborhood, giving only small weight to far away points that may belong to a different cluster. Specifically we choose

$$\mathbf{X}_i^{new} = (1-\alpha)\mathbf{X}_i^{old} + \sum_{\substack{j=1 \\ i \neq j}}^{N} w_i(\mathbf{X}_i, \mathbf{X}_j)\mathbf{X}_j^{old}. \tag{6}$$

This smoothing process acts on each component as follows

$$(x_i^a)^{new} = (1-\alpha)(x_i^a)^{old} + \sum_{\substack{j=1 \\ i \neq j}}^{N} w_i(\mathbf{X}_i, \mathbf{X}_j)(x_j^a)^{old} \quad a = 1, \cdots, d. \tag{7}$$

This averaging process is similar, for each component, to the scalar process Eq. 2 with the notable difference that the weights depend on the distance between the points in $\mathbb{R}^d$. This means that the averages of the components are not independent. We take for example the Euclidean distance between points:

$$d(\mathbf{X}_i, \mathbf{X}_j) = \sqrt{\sum_{a=1}^{d} (x_i^a - x_j^a)^2},$$

and choose a weight w_i as a function of this distance

$$w_i(\mathbf{X}_i, \mathbf{X}_j) = \frac{\beta_i}{1 + d(\mathbf{X}_i, \mathbf{X}_j)^\gamma} \qquad i \neq j$$

with

$$\beta_i = \alpha \left(\sum_{\substack{j=1 \\ j \neq i}}^{N} \frac{1}{1 + d(\mathbf{X}_i, \mathbf{X}_j)^\gamma} \right)^{-1} .$$

Such a choice will make nearby points more "influential" on the displacement of x_i.

The distance that involve all the components of a point provides the coupling in Eqs. 7 between the averaging processes of the different components. See Figure 2.

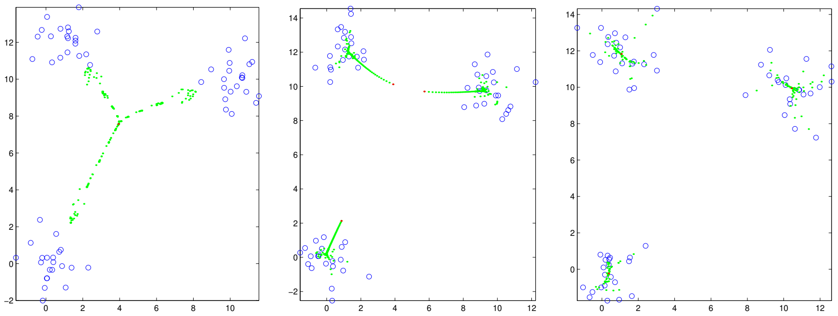

Figure 2. Random vectors (points in $\mathbb{R}^2$) averaging process with $\gamma = 0.5, 1.2, 2$ (left to right)

2.3. Averaging Points on a Circle (values in $\mathbf{S}^1$)

Suppose next that we have a set of values on the unit circle $\mathbf{S}^1$, encoded say as complex numbers via $\{e^{i\theta_j}\}$ where $\theta_j \in [0, 2\pi)$. We would like to devise a process of averaging over the values on $\mathbf{S}^1$ that can effectively summarize for us the initial set of values. It is clear that we cannot simply average the complex numbers directly. Such an operation will most likely result in a value not on the unit circle. An alternative is to average the angles $\{\theta_j\}$ as numbers between $[0, 2\pi]$, however, we immediately realize that $\theta_1 = \epsilon$ and $\theta_2 = 2\pi - \epsilon$ will yield an average of π, far away from both ϵ, and $2\pi - \epsilon$ on the circle. The proper averaging in this case should obviously be $\theta_{av} = 0$, and certainly not π. The problem here arises from the "jump" in our mapping of $\mathbf{S}^1$ into the representation space Θ of the angles. Clearly the "average" of θ_1 and θ_2 should be the angle that bisects the shortest arc that connects θ_1 to θ_2. Suppose we have N points on the unit circle and we wish to devise a process of averaging that will yield a better representation of these points (e.g. via clustering them). We shall map these points $P_1(\theta_1), P_2(\theta_2) \cdots P_N(\theta_N)$ into multiple images on a real line (see Fig. 3) exhibiting the fact that P_j corresponds to $\{\theta_j + k \cdot 2\pi\}$, $k = 0, \pm 1, \pm 2 \ldots$ Hence, we have a periodic configuration of points on R that exhibits (maps) the single angle parameter.

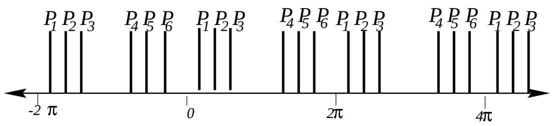

Figure 3. Averaging $\mathbf{S}^1$ values.

Now, the (angular)distance between two points on $\mathbb{R}$ can be simply read as a distance on $\mathbb{R}$: $d(P_1, P_2)$ being defined as the *closest distance* on $\mathbb{R}$ between the multiple representations of P_1 and P_2. This corresponds to always measuring the distance on $\mathbf{S}^1$ by the smallest arc connecting P_1 and P_2 (as is natural). The averaging (smoothing) process for these points can now be defined as follows:
For all P_i look at the points in its symmetric 2π-neighborhood, i.e. consider the interval $(P_i - \pi, P_i + \pi)$. Then consider all points in this neighborhood as Q_j's. Now compute

$$P_i^{\text{next}} = (1 - \alpha) P_i + \alpha \sum_{j \neq i} w(d(P_i, Q_j)) Q_j$$

Clearly this process will move P_i a bit toward the weighted average of the "influential" parts $\{Q_j\}$, with the influence function defined to depend on the distance between P_i and Q_j.

It is clear from the above defined process that periodicity is here preserved, hence we shall always get "a correct" representation of N points on a circle, and the various possibilities for choosing influence functions $w()$ will lead to a variety of types of clustering processes. It is also clear that, although the averaging process defined above works on infinitely many representations of each point, its practical implementation can easily be done by an updating process of only N

points. It was simply a *conceptual advantage* to look at the problem on $\mathbb{R}$ rather than work directly on $\mathbf{S}^1$. Note that the distance defined this way is the geodesic distance on the circle.

2.4. Adaptively Smoothing a Discrete Time Series

Suppose that we are given a discrete time series $\mathbf{X} = (x_1, x_2, \cdots, x_N)^T$. Here $x_i = x(t_i)$ and this fact makes this series different from the series of scalars we discussed above. Sets of this type appear often in almost every field of research ranging from Engineering and Physics through Biology and Psychology to Economy and Sociology. Our adaptive averaging is based, as above, on the distance function $d(x_i, x_j)$:

$$x_i^{n+1} = (1-\alpha)x_i^n + \sum_{j \neq i} w_i(d(x_i^n, x_j^n))x_j^n,$$

where the superscript n is the iterations index.

In order to go from this principle to a specific algorithm we have to specify the "smoothing function" w and more important we have to decide what is the "appropriate distance" between generic two measurements x_i and x_j.

We may choose a smoothing function

$$w_i(x_i, x_j) = \frac{\beta_i}{1 + d(x_i, x_j)^\gamma} \qquad i \neq j. \tag{8}$$

where γ defines the degree of smoothing wanted.

The distance $d(x_i, x_j)$ could be $|t_i - t_j|$ or it could be $\alpha|t_i - t_j| + \beta|x_i - x_j|$ or it could even be distance measured *on the signal curve.* For two adjacent measurements x_i and x_{i+1} the distance on the signal is

$$d_i^2(x_i, x_{i+1}) = |t_{i+1} - t_i|^2 + |x_{i+1} - x_i|^2.$$

Note that the index i indicates that this distance is between two adjacent points. The distance between generic points is the sum of the distances between the pairs of adjacent points that connect the given generic points. Assuming, with no loss of generality, that $j > i$ it reads

$$d(x_i, x_j) = \sum_{k=i}^{j-1} d_k(x_k, x_{k+1}).$$

This expression is simply an approximation of the length of the curve which connects x_i and x_j.

2.5. Adaptively Smoothing an Image

An image is a two-dimensional analog of the discrete time series. Denote the gray value at a pixel $I_{ij} = I(x_i, y_j)$ and we assume equal spacing in the x and y directions. Denoising the image will clearly necessitate a smoothing process. The special nature of images demands the average to be taken over pixels that "resemble" the center pixel I_{ij}. This average should be taken over projections of the same object and be independent of pixels that describe different objects or that are simply in a region far from the pixel of interest.

The smoothing step can therefore be taken as

$$I_{ij}^{n+1} = (1-\alpha)I_{ij}^n + \sum_{kl \neq ij} w_{ij}(d(I_{ij}^n, I_{kl}^n))I_{kl}^n.$$

where, again

$$w_{ij}(I_{ij}, I_{kl}) = \frac{\beta_i}{1 + d(I_{ij}, I_{kl})^\gamma} \qquad (ij) \neq (kl). \tag{9}$$

However, here we represent the image as a two-dimensional surface which is the discretized graph of $I(x, y)$. The distance is measured *on the image surface* as follows. For any pixel the distance to pixels in the closest neighborhood, i.e. 3x3 stencil, is calculated as

$$d_{near}^2(I_{ij}, I_{kl}) = |i-k|^2 + |j-l|^2 + |I_{ij} - I_{kl}|^2.$$

For further away points the distance is defined as sum of distances of near neighbors along the shortest path *on the manifold* that connects the two points. Recently, the fast marching method on curved domains algorithm was developed in (Kimmel and Sethian, 1998), it can compute these geodesic distances very effectively. See, Fig. 4 for adaptive smoothing with various γ values.

3. Short Time Kernel for Non-Linear Differential Averaging

The solution of the linear diffusion equation is given as a convolution with a Gaussian. The Gaussian function with a time dependent variance is called the *kernel* of the differential equation. For a non-linear diffusion the situation is different and a global (in time) kernel does not exist. If, however, we are interested in the evolution over a short time span, more can be said. A kernel can be constructed for the short time evolution of the non-linear diffusion equation (Cohen, Hagin and Keller, 1972). We demonstrate this and clarify the ideas that lie at the basis of any of the geometrical evolution equations in the rest of this section.

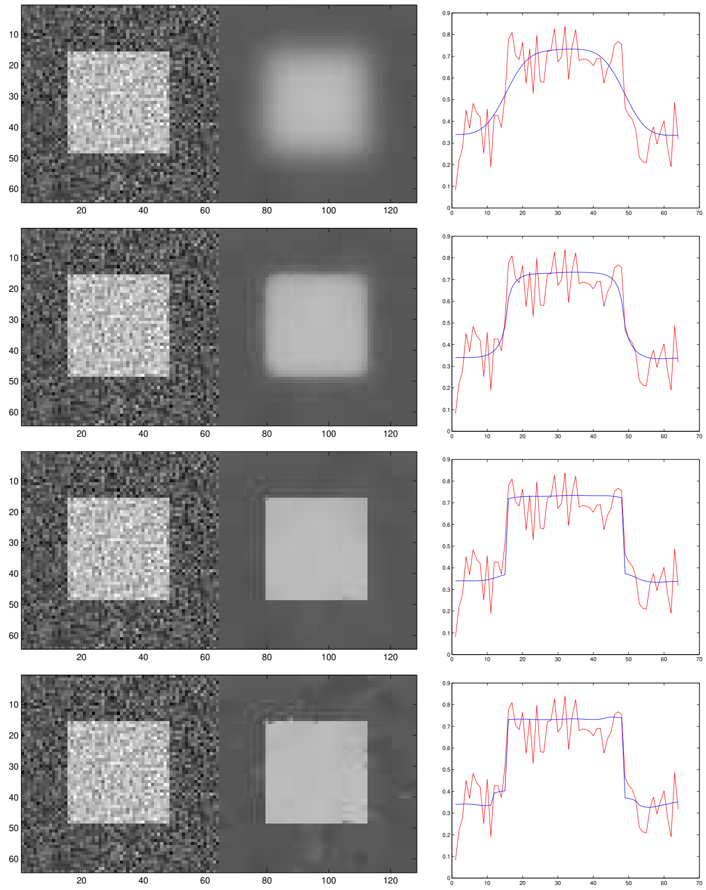

Figure 4. Image adaptive averaging with $\gamma = 0, 1, 1.5, 2$ (top down), with a cross-section that shows the edge preserving property.

3.1. One-dimensional signal

Let us first consider a one dimensional signal $Y(p)$. The signal can be thought of as a curve embedded in $\mathbb{R}^2$, as $(X^1(p), X^2(p)) = (p, Y(p))$ see Fig. 5. The equation we analyze is the geometric heat equation

$$Y_t(p) = Y_{ss}(p), \tag{10}$$

where s is an arclength defined in terms of a general metric by the relation

$$ds^2 = g(p)dp^2,$$

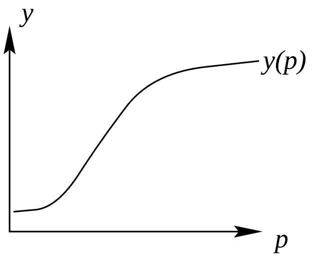

Figure 5. The signal as a curve

and for any function f we denote by f_s the derivative $\partial f/\partial s = \frac{1}{\sqrt{g(p)}}\partial f/\partial p$. This is the gradient descent equation for the Polyakov action

$$S[Y,g] = \int dp\sqrt{g}g^{-1}\left((X_p^1)^2 + (X_p^2)^2\right) = \int ds(X_s^2 + Y_s^2)$$

If we further assume that the metric on the curve is induced from the ambient Euclidean $\mathbb{R}^2$ space then $g(p) = 1+Y_p^2$ and $g(s) = X_s^2+Y_s^2 = 1$ from the definition of the arclength s. The Polyakov functional has a very simple geometric meaning in this case:

$$S[Y] = \int ds(X_s^2 + Y_s^2) = \int ds = \text{length}.$$

Note however that that other metrics are possible, e.g the L_γ-norm

$$g(p) = \left(\left(\frac{\partial X}{\partial p}\right)^\gamma + \left(\frac{\partial Y}{\partial p}\right)^\gamma\right)^{\frac{2}{\gamma}}.$$

Note that these functions are metrics since they transform properly under a general reparameterization $p \to p(\tilde{p})$. In our canonical coordinate system $(p, Y(p)$ the metric admit the form

$$g(p) = \left(1 + \left(\frac{\partial Y}{\partial p}\right)^\gamma\right)^{\frac{2}{\gamma}}.$$

Use of these metrics for the definition of a distance leads, upon discretization, to the smoothing function $w()$ of the previous section.

From an axiomatic point of view we may also wish to construct a curve flow which is invariant under different groups of transformations. Invariance under the Euclidean group that includes translations and rotations leads to the Euclidean induced metric, and results in a curvature flow projected on the Y axis. This projection action preserves edges longer along the flow, enhancing this way the boundary between different regions in the image as will be seen below.

Different requirements (such as invariance under different groups of transformations) lead to a different form for the metric and to a different flow. We analyze all these possibilities at the same time by not specifying any particular form of the metric and leaving it as a free "parameter" of the framework.

The flow is the one-dimensional analog of the Beltrami flow

$$Y_t = \frac{1}{\sqrt{g}}\partial_p(\sqrt{g}g^{-1}\partial_p Y) \equiv \Delta_g Y \tag{11}$$

where g is the metric on the curve. Another way to write this equation is

$$Y_t = D_p\partial_p Y = (\partial_p + A')g^{-1}\partial_p Y = g^{-1}(\partial_p^2 Y + A\partial_p Y) \tag{12}$$

where $D_p = \partial_p + A'$ is the covariant derivative, $A' = \frac{1}{2}g^{-1}\partial_p g$ is the connection and $A = A' - g^{-1}\partial_p g = -\frac{1}{2}g^{-1}\partial_p g$. This equation can describe a variety of curve evolution dynamics upon different choices of the metric g. In more complicated situations we encounter the same form of Eq. 12 where the connection A depends on the metric of the embedding space as well. In flows with g depending on $\partial_p Y$ Eq. 12 is non-linear, and prevents the existence of kernels for long time intervals. If however $g \equiv 1$ then eq. 11 becomes the usual heat equation with the well known Gaussian smoothing kernel.

Upon discretization $Y(p) \to Y_i$ this equation assumes the following form

$$Y_i^{n+1} = \sum_j W_{ij}^n Y_j^n$$

where n is the iteration index and W^n is a matrix whose entries depend on the values of Y^n and $\partial_p Y^n$. For one iteration only we can think about the update rule as if W^n is a fixed function of p describing the underlying curve which is **fixed** during this one short time update. After Y is updated the metric is updated and then again it is fixed during the next update of Y. The coupling between the metric (or connection), that describes the geometry, and the feature of interest Y, prevents a global fundamental solution to exist, yet we can find such solution for each time step in which the geometry is fixed.

In order to derive the *short time kernel* of these equations we use the following ansatz, known in physics as the WKB approximation,

$$Y(p, t+\epsilon) = \int K(p, p'; \epsilon)Y(p', t)dp', \tag{13}$$

where the kernel is assumed to be of the form

$$K(p, p'; t) = \frac{H(p, p', t)}{\sqrt{t}}\exp\{-\frac{\Psi(p, p')}{t}\}. \tag{14}$$

We take, without loss of generality, $H(p, p', t) = \text{constant}$, see Appendix A for the details. This form is a generalization of the Gaussian kernel solution of the linear diffusion equation. The function Ψ depends on the diffusion tensor only while H depends on the details of all the terms of the equation. The validity of this approximation procedure can be found in (Cohen, Hagin and Keller, 1972) for example.

Note that upon discretization of the curve $p = ah$, where $h = L/N$, L is the length of the curve, and N is the number of segments, this equation takes the form of a system of linear equations

$$X_a^{i\ new} = \sum_{b=1}^{N} K_{ab} X_b^{i\ old}. \tag{15}$$

This has the same form as the one described in the previous section when we discussed averaging sequences of scalars and vectors.

Inserting Eq. (13) in Eq. (11) we see that the kernel $K(p, p'; t)$ as a function of p satisfies the same equation as X^i. As a power series in t we get

$$\left(\frac{\Psi(p,p')}{t^2} + O(\frac{1}{t})\right) K(p,p';t) = \left(\frac{1}{t^2} g^{-1}(p)(\partial_p \Psi(p,p'))^2 + O(\frac{1}{t})\right) K(p,p';t). \tag{16}$$

For short times only the most singular part is dominant, namely

$$\Psi(p) = g^{-1}(p)(\partial_p \Psi(p))^2, \tag{17}$$

where by abuse of notations $\Psi(p)$ is a shorthand for $\Psi(p, p')$, and with boundary condition

$$\Psi(p') = 0. \tag{18}$$

This equation can be rewritten as an algebraic–differential system of equations

$$\begin{aligned} F(p, \Psi, Z) &= Z^2 - g(p)\Psi = 0 \\ Z &= \frac{\partial \Psi}{\partial p} \end{aligned} \tag{19}$$

The algebraic equation is solved to yield

$$Z = \sqrt{g\Psi} \tag{20}$$

The resulting differential equation is solved by separation of variables

$$\frac{d\Psi}{\sqrt{\Psi}} = \sqrt{g(p)}dp, \tag{21}$$

that yields the solution

$$\Psi(p) = \frac{1}{4}\left(\int_p^{p'} \sqrt{g}\, d\tilde{p}\right)^2 = \frac{1}{4}\left(\int_p^{p'} ds\right)^2 . \tag{22}$$

This equation has a simple interpretation. The kernel is a Gaussian

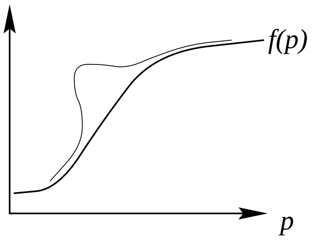

Figure 6. Gaussian on the curve

on the curve, i.e. the convolution is performed on the signal (see Fig. 6). Distances from a point should be measured on the signal itself and not on the grid that happens to be used in order to describe it. This way, points on one side of a significant jump are farther away from the points on the other side of the jump and therefore have small influence on the average there. This explains the edge preserving nature of the geometric heat equation and the filter that emerges out of it.

In Figure 7 a one dimensional 'edge' is convolved with a Gaussian kernel once with support along the x axis, and once, intrinsically defined, i.e., with a support of the signal itself. The variance in both cases was defined to yield similar results along the flat areas. We see that the kernel defined on the signal better preserves the bimodal nature of the data.

3.2. One-dimensional curve embedded in $\mathbb{R}^n$

Equation (11) can be easily generalized to an arbitrary one-dimensional curve embedded in $\mathbb{R}^n$ by simply applying the geometric diffusion equation componentwise:

$$X^i_t = \frac{1}{\sqrt{g}}\partial_p(\sqrt{g}g^{-1}\partial_p X^i) \equiv \Delta_g X^i \qquad i = 1, 2, \ldots, n. \tag{23}$$

This equation can describe a variety of curve evolution dynamics upon different choices of the metric g. Note that the metric involves all components leading to a system of coupled equations.

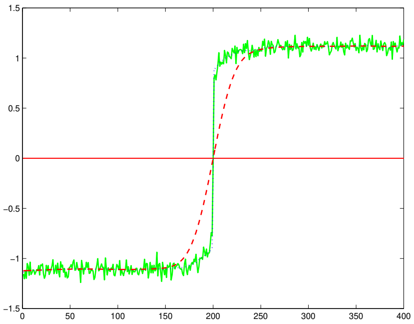

Figure 7. Smoothing a noisy signal (green solid line) once with a Gaussian along the p axis (red dashed line) and with the kernel defined on the signal itself (blue doted line). For a better view see http://www.cs.technion.ac.il/~ron/pub.html

In order to derive the short time kernel of these equations we use the following ansatz

$$X^i(p, t+\epsilon) = \int K(p, p'; \epsilon) X^i(p', t) dp' \qquad (24)$$

where the kernel form, like in Eq. (14) is given by

$$k(p, p'; t) = \frac{C}{\sqrt{t}} \exp\{-\frac{\Psi(p, p')}{t}\} \qquad (25)$$

Now, C is a constant and Ψ is the same as in the previous subsection, i.e.,

$$\Psi(p) = \frac{1}{4}\left(\int_p^{p'} \sqrt{g} dp''\right)^2 = \frac{1}{4}\left(\int_p^{p'} ds\right)^2 = \frac{1}{4} s(p, p')^2. \qquad (26)$$

$s(p, p')$ is the distance on the curve between point p and the point p'.

3.3. Affine invariant Averaging

Once we have a formulation for the kernel in terms of an arc-length through the relation $ds^2 = g dp^2$, we can envision the construction of group invariant kernels by imposing invariance properties on the metric. The simplest example is the special affine invariant metric of a curve, i.e.

$$s(p_0, p') = \int_{p_0}^{p'} \langle \mathbf{X}_p, \mathbf{X}_{pp} \rangle^{1/3} dp. \qquad (27)$$

See e.g. (Sapiro and Tannenbaum, 1998; Alvarez et al., 1993; Calabi et al., 1998).

3.4. Relation to other works

3.4.1. *The bilateral filter*

The short time kernel can be approximated in a small window by approximating the integral

$$\int_{p_0}^{p_0+dp} \sqrt{g(p')}dp' = \sqrt{g(p)}dp.$$

We can use the Euclidean induced metric, i.e.,

$$g = 1 + (\frac{\partial f}{\partial p})^2.$$

Then,

$$4\Psi = (\sqrt{g(p)}dp)^2 = (1 + (\frac{\partial f}{\partial p})^2)dp^2 = dp^2 + df^2,$$

which is exactly the bilateral filter that was recently proposed by Tomasi and Manduchi, (Tomasi and Manduchi, 1998; Elad and Shaked, 1999; Barash, 1999).

3.4.2. *The TV digital filter*

Recently Chan, Osher and Shen(Chan, 2000) analyzed non-linear filter of the following form

$$U^{new} = \sum_{\beta\sim\alpha} h_{\alpha\beta} U^{old}_{\beta},$$

where the sum is over neighbors of α (including α itself). The weights $h_{\alpha\beta}$ are of the form

$$h_{\alpha\beta} = \frac{w_{\alpha\beta}}{\lambda + \sum_{\gamma\sim\alpha} w_{\alpha\gamma}}$$

for $\alpha \neq \beta$. The diagonal weight is

$$h_{\alpha\alpha} = \frac{\lambda}{\lambda + \sum_{\gamma\sim\alpha} w_{\alpha\gamma}}.$$

These filters are constructed by direct discretization of the differential operator that results in a gradient descent minimization of the TV functional. In practice it is a regularized functional that is minimized leading to a direct relation to the Beltrami flow which is based on the Euclidean induced metric (see (Sochen, Kimmel and Malladi, 1998) for details). Writing the discretized differential operator as a matrix whose entries depend on the signal appear also in (Weickert, 1998; Vogel and Oman, 1996).

The short time approach is different since we analyze and eventually discretize *the solution* of the differential equation. It is also more general since we treat a variety of flows by not specifying the explicit form of the metric at the outset. In this way we have a clear intuitive understanding of the adaptive averaging as a Gaussian weight function on the manifold that is defined by the data. The flows differ in the geometry attributed to the manifold through different choices of the metric.

3.4.3. *Relation to Projective Averaging*

In (Bruckstein and Shaked, 1997) an adaptive non-linear and projective invariant averaging process was introduced. It was demonstrated that it is equivalent to a projective invariant PDE that generates the curve flow.

4. Averaging Constrained Features

One often encounter, in some signal and image processing tasks, a more complex feature-space signal. Examples of such features are color, texture, curvature, derivative vector field, orientation vector field etc. The feature space may have non-trivial geometry represented by a metric (Sochen, 1999; Kimmel Malladi and Sochen, 2000; Chan and Shen, 1999; Tang, Sapiro and Caselles, 1999; Kimmel and Sochen, 2000). Regularization or smoothing of the feature space is frequently done with a non-linear diffusion system of equations. In this section we derive the short time kernel for the diffusion of an orientation defined over a one-dimensional curve.

4.1. Orientation field over a signal

In the case of averaging an orientation field, we define the orientation at each point $(p, Y(p))$ as the angle between the feature vector and the p axis. By definition, the vector field has a unit magnitude and this property should be preserved during the smoothing flow.

The vector field is a map $f : C \to \mathcal{S}^{\infty}$, and the geometric smoothing flow may here too be derived as a gradient descent flow from the Polyakov action defined as

$$f_t(p) = \Delta_g f(p) + \Gamma(\partial_p f)^2 g^{-1}, \tag{28}$$

see (Kimmel and Sochen, 2000), where

$$\Gamma = h^{-1}\partial_f h, \tag{29}$$

and h is the metric on $\mathbf{S}^1$. Thus, for example, if the components of the orientation vector field are f and e such that $f^2+e^2=1$, then the metric on the patch that is described by f is calculated as follows

$$ds^2 = df^2 + de^2 = df^2 + (\frac{\partial e}{\partial f}df)^2 = (1+\frac{f^2}{1-f^2})df^2 = \frac{1}{1-f^2}df^2.$$

From the definition

$$ds^2 = hdf^2,$$

we get

$$h = \frac{1}{1-f^2}.$$

The calculation of Γ is straightforward and gives

$$\Gamma = h^{-1}\partial_f h = (1-f^2)\partial_f(1-f^2)^{-1} = \frac{f}{1-f^2} = fh.$$

Let us write Eq. 28 in a more suggestive form:

$$f_t = \left(\partial_p + \frac{1}{2}g^{-1}\partial_p g + \Gamma\partial_p f\right) g^{-1}\partial_p f. \tag{30}$$

We simplify this expression by noting that

$$\Gamma\partial_p f = \frac{1}{2}h^{-1}\partial_f h \partial_p f = \frac{1}{2}h^{-1}\partial_p h.$$

Finally we get

$$f_t = (\partial_p + A')\, g^{-1}\partial_p f = g^{-1}\,(\partial_p + A)\,\partial_p f$$

where $A' = \frac{1}{2}g^{-1}\partial_p g + \frac{1}{2}h^{-1}\partial_p h$ and $A = -\frac{1}{2}g^{-1}\partial_p g + \frac{1}{2}h^{-1}\partial_p h$.

Repeating the analysis of the previous subsections we derive the solution of the form

$$K(p,p';t) = \frac{H(p,p',t)}{\sqrt{t}}\exp\{-\frac{\Psi(p,p')}{t}\}. \tag{31}$$

and develop $H(p,p';t)$ in a Taylor series

$$H(p,p';t) = \sum_{n=0}^{\infty} t^n H_n(p,p').$$

The leading term is the same as in the non-constrained diffusion and therefore

$$\Psi(p) = \frac{1}{4}\left(\int_p^{p'}\sqrt{g}dp''\right)^2 = \frac{1}{4}\left(\int_p^{p'} ds\right)^2. \tag{32}$$

Here g is the induced metric

$$g = 1 + (Y_p)^2 + h(\partial_p f)^2$$

and while in the previous case H = constant here the situation is different and a careful analysis (see Appendix) gives

$$H = Ch^{-1/4} + O(t) \tag{33}$$

where C is a constant and h is the metric on $\mathbf{S}^1$.

4.2. Orientation field over a curve embedded in $\mathbb{R}^n$

The above analysis applies to a constrained vector field defined over a curve embedded in $\mathbb{R}^n$. The map is $f : C \to \mathbf{S}^1$ and the PDE associated with it is

$$f_t(p) = \Delta_g f(p) + \Gamma(\partial_p f)^2 g^{-1} \tag{34}$$

The only difference is in the structure of the inner metric g.

$$g = \sum_i (X_p^i)^2 + h(\partial_p f)^2 = \sum_i (X_p^i)^2 + \frac{1}{1-f^2}(\sum_i \frac{\partial f}{\partial X^i} X_p^i)^2$$

The short time kernel is the same as in the above subsection.

It is important to understand that these equations are of the same form for any coordinate system we choose (up to transformation of coordinates of course). We need at least two charts to cover $\mathbf{S}^1$ but we can also adopt the ideas developed in Section 2 and construct for each point the coordinate system where the orientation is the furthest point from the singularity.

5. Image Denoising Results

The short time kernel for images is conceptually a straitforward generalization of the 1D case. It is given as a weighted Gaussian on the image manifold (see (Sochen, 1999) for detailes):

$$K(p, p') = \frac{C}{t} e^{-\frac{d(p,p')^2}{t}} + O(t^0)$$

Here p and p' are two points on the image manifold and $d(p, p')$ is the distance between them i.e. the length of the shortest geodesic on the manifold between these points. The results for small window are given below in Figure 8.

Figure 8. In each couple the left is the original image, while the right is the filtered one processed with the Beltrami approximated by the bilateral filter. The convolution kernel is $e^{-ds^2} = e^{-(dx^2+dy^2+0.1dI^2)/16}$, normalized, with window size of 5×5.

6. Summary and Conclusions

We briefly explored here the relation between PDE based filters, classical signal processing linear filters, and non-linear filters. It was shown that the short time kernels of the Beltrami flow may be considered as approximations for known linear and non-linear filters. The discussion covers Gaussian filters, the non-linear bilateral filters, and further non-trivial robust and data-adaptive filters. Our approach yields a unified and comprehensive view on the relation between these seemingly unrelated set of tools.

Acknowledgements

Stimulating discussions with Michael Elad of Net2Wireless, Doron Shaked and Danny Barash of HP Labs. on bilateral filters and the Beltrami flow, are gratefully acknowledged. We thank Zeev Schuss for discussions and for pointing us to very valuable references.

Appendix

We analyze in this appendix the time structure of a general WKB approximation. The Kernel obeys the equation

$$K_t = g^{-1}(\partial_p + A)\partial_p K \tag{35}$$

and we define a WKB kernel in the form

$$K(p,p';t) = \frac{H(p,p',t)}{\sqrt{t}} \exp\{-\frac{\Psi(p,p')}{t}\}. \tag{36}$$

and develop $H(p,p';t)$ in a Taylor series

$$H(p,p';t) = \sum_{n=0}^{\infty} t^n H_n(p,p')$$

Inserting this ansatz in the diffusion equation 35 we get on one hand

$$\frac{\partial K}{\partial t} = \frac{\Psi}{t^2}\frac{H(p,p',t)}{\sqrt{t}} \exp\{-\frac{\Psi(p,p')}{t}\} + \sum_{n=0}^{\infty}(n-\frac{1}{2})t^{n-\frac{3}{2}} H_n(p,p') \exp\{-\frac{\Psi(p,p')}{t}\}.$$

On the other hand we find

$$\partial_p K = -\frac{H(p,p';t)\partial_p \Psi}{t^{3/2}} \exp\{-\frac{\Psi(p,p')}{t}\} + \frac{\partial_p H}{t^{1/2}} \exp\{-\frac{\Psi(p,p')}{t}\}$$

and

$$\begin{aligned}\partial_p^2 K &= -\frac{H(p,p';t)\partial_p^2 \Psi}{t^{3/2}} \exp\{-\frac{\Psi(p,p')}{t}\} - 2\frac{(\partial_p H)(\partial_p \Psi)}{t^{3/2}} \exp\{-\frac{\Psi(p,p')}{t}\} \\ &+ \frac{H(p,p';t)(\partial_p\Psi)^2}{t^{5/2}} \exp\{-\frac{\Psi(p,p')}{t}\} + \frac{\partial_p^2 H(p,p';t)}{t^{1/2}} \exp\{-\frac{\Psi(p,p')}{t}\}\end{aligned}$$

By comparison of the coefficients of the power series we get from the leading term

$$\Psi H_0 = g^{-1}(\partial_p \Psi)^2 H_0$$

and from the second term

$$\Psi H_1 - \frac{1}{2}H_0 = g^{-1}\left(H_1(\partial_p\Psi)^2 - 2(\partial_p\Psi)(\partial_p H_0) - h_0(\partial_p + A)\partial_p\Psi\right) \tag{37}$$

The coefficient of H_1 is the equation for the leading term and thus assumed to be satisfied. For the rest, we know from the solution of the first equation that

$$\partial_p\Psi = \sqrt{g\Psi}$$

and

$$\partial_p^2\Psi = \frac{(\partial g)\Psi + g\partial\Psi}{2\sqrt{g\Psi}} = \frac{1}{2}\sqrt{\frac{\psi}{g}}\partial_p g + \frac{g\sqrt{g\Psi}}{2\sqrt{g\Psi}} = \frac{1}{2}\sqrt{\frac{\psi}{g}}\partial_p g + \frac{g}{2}.$$

Using these identities we get

$$\frac{g}{2}H_0 = H_0(\partial_p^2\Psi + A\partial_p\Psi) + 2(\partial_p\Psi)(\partial_p H_0) = H_0\left(\frac{1}{2}\sqrt{\frac{\Psi}{g}}\partial_p g + \frac{1}{2}g + A\sqrt{g\Psi}\right) + 2\sqrt{g\Psi}(\partial_p H_0)$$

from which we finally obtain

$$\partial_p H_0 + (\frac{1}{4}g^{-1}\partial_p g + \frac{1}{2}A)H_0 = 0.$$

The form of A when the feature space is not constrained is

$$A = \sqrt{g}\partial_p\frac{1}{\sqrt{g}} = -\frac{1}{2}g^{-1}\partial_p g$$

We conclude that in this case $H_0 =$ constant. In the constrained case $A = -\frac{1}{2}g^{-1}\partial_p g + \frac{1}{2}h^{-1}\partial_p h$. The coefficient H_0 is in this case

$$H_0 = Ch^{-1/4}$$

where C is a constant of integration.

References

Alvarez L., F. Guichard, P. L. Lions, and J. M. Morel, "Axioms and fundamental equations of image processing". *Arch. Rational Mechanics*, 123, 1993.

Barash D. "Bilateral filtering and anisotropic diffusion: towards a unified viewpoint", Technical Report HPL-18-2000, HP Labs., 2000.

Black M., G. Sapiro, D. Marimont, and D. Heeger, "Robust anisotropic diffusion", *IEEE Trans. on Image Processing*, 7(3), 1998.

Bruckstein A. M. and D. Shaked. "On projective invariant smoothing and evolutions of planar curves and polygons". *Journal of Mathematical Imaging and Vision*, 7:225-240, 1997.

Calabi, E., P.J. Olver, C. Shakiban, A. Tannenbaum, and S. Haker, "Differential and numerically invariant signature curves applied to object recognition', *Int. J. Computer Vision* 26:107-135, 1998.

Comaniciu D. and P. Meer, "Mean shift analysis and applications" *Proc. of the 7th IEEE Int. Conf. on Computer Vision* CA, USA, 2:1197-203, 1999

Chan F. C., Osher S. and Shen J. "The digital TV filter and nonlinear denoising". *UCLA-Tecnical report, 2000.*

Chan T. and J. Shen, Variational restoration of non-flat image features: Models and algorithms" Technical report, Math-UCLA, 1999.

Cohen J. K., Hagin F. G. and Keller J. B. "Short time asymptotic Expansions of solutions of parabolic equations" *Journal of Mathematical Analysis and Applications*, 38:82-91, 1972.

Elad M. and Shaked D. Personal communication. In *HP Labs Israel*, 1999.

Geman D. and Reynolds G., *IEEE Trans. on PAMI*, 14:376-383, 1992.

Kimmel R., Malladi R. and Sochen N. "Images as embedded maps and minimal surfaces: movies, color, texture, and volumetric medical images". *International Journal of Computer Vision,* in press, 2000.

Kimmel R., and J. A. Sethian, "Computing Geodesic Paths on Manifolds", *Proceedings of National Academy of Sciences, USA*, 95(15):8431-8435, 1998.

Kimmel R. and Sochen N. "Orientation diffusion or how to comb a porcupine ?",. *Journal of Visual Communication and Image Representation*, in press 2000.

Mumford D. and J. Shah. "Boundary detection by minimizing functionals". In *Proc. of CVPR, Computer Vision and Pattern Recognition*, San Francisco, 1985.

Perona P. "Orientation diffusions". *IEEE Trans. on Image Processing*, 7(3):457-467, 1998.

Perona P. and J. Malik. "Scale-space and edge detection using anisotropic diffusion". *IEEE-PAMI*, 12:629–639, 1990.

Rudin L., S. Osher, and E. Fatemi. "Nonlinear total variation based noise removal algorithms". *Physica D*, 60:259–268, 1992.

Sapiro S., and A. Tannenbaum, "Affine invariant scale space", *Int. Journal of Computer Vision*, 11(1):25-44, 1993.

Sochen N., R. Kimmel and R. Malladi, "A geometrical framework for low level vision", *IEEE Trans. on Image Processing*, 7(3):310-318, 1998.

Sochen N., "Stochastic processes in vision I: From Langevin to Beltrami," *CC Pub #285* June 1999, Technion, Israel.

Tang B., G. Sapiro and V. Caselles "Direction diffusion", International Conference on Computer Vision 1999.

Tomasi C. , and R. Manduchi, "Bilateral filtering for gray and color images". In *Proc. of the IEEE International Conference on Computer Vision*, 839-846, 1998.

Vogel R. and M. E. Oman, "Iterative methods for total variation denoising", *SIAM. J. Sci. Statist. Comput.*, 1996.

J Weickert. *Anisotropic Diffusion in Image Processing.* Teubner Stuttgart, 1998. ISBN 3-519-02606-6.