

Preliminaries

A NUMBER of existing algorithms for image restoration and segmentation, which are related to SIDEs, are summarized in this chapter.

■ 2.1 Notation.

In this section, we describe the notation which is used in the current chapter. Most of this notation will carry over to the rest of the thesis; however, the large quantity of symbols needed will force us to adopt a slightly different notation in Chapter 4—which we will describe explicitly in Section 4.2.

We begin with the one-dimensional (1-D) case. The 1-D signal to be processed is denoted by $u^0(x)$. The superscript 0 is a reminder of the fact that the signal is to be processed via a partial differential equation (PDE) of the following form:

$$\begin{aligned} u_t &= \mathcal{A}_1(u, u_x, u_{xx}) \\ u(0, x) &= u^0(x). \end{aligned} \tag{2.1}$$

The variable t is called *scale* or *time*, and the solution $u(t, x)$ to (2.1), for $0 \leq t < \infty$, is called a *scale space*. The partial derivatives with respect to t and x are denoted by subscripts, and \mathcal{A}_1 is an operator. The scale space is called *linear* (*nonlinear*) if \mathcal{A}_1 is a linear (nonlinear) operator.

Similarly, an image $u^0(x, y)$ depending on two spatial variables, x and y , will be processed using a PDE of the form

$$\begin{aligned} u_t &= \mathcal{A}_2(u, u_x, u_y, u_{xx}, u_{yy}, u_{xy}) \\ u(0, x, y) &= u^0(x, y), \end{aligned} \tag{2.2}$$

which generates the scale space $u(t, x, y)$, for $0 \leq t < \infty$. In the PDEs we consider, the right-hand side will sometimes involve the *gradient* and *divergence* operators. The gradient of $u(t, x, y)$ is the two-vector consisting of the partial derivatives of u with respect to the spatial variables x and y :

$$\nabla u \stackrel{\text{def}}{=} (u_x, u_y)^T, \tag{2.3}$$

where the superscript T denotes the transpose of a vector. The norm of the gradient is:

$$|\nabla u| \stackrel{\text{def}}{=} \sqrt{u_x^2 + u_y^2} \quad (2.4)$$

The divergence of a vector function $(u(x, y), v(x, y))^T$ is:

$$\vec{\nabla} \cdot \begin{pmatrix} u \\ v \end{pmatrix} \stackrel{\text{def}}{=} u_x + v_y. \quad (2.5)$$

We also consider semi-discrete versions of (2.1) and (2.2), obtained by discretizing the spatial variables and leaving t continuous. Specifically, an N -point 1-D discrete signal to be processed is denoted by \mathbf{u}^0 ; it is an element of the N -dimensional vector space \mathbb{R}^N . We exclusively reserve **boldface** letters for vectors—i.e., discrete signals and images. The vector \mathbf{u}^0 is the initial condition to the following N -dimensional ordinary differential equation (ODE):

$$\begin{aligned} \dot{\mathbf{u}}(t) &= \mathcal{B}_1(\mathbf{u}(t)) \\ \mathbf{u}(0) &= \mathbf{u}^0, \end{aligned} \quad (2.6)$$

where $\mathbf{u}(t)$ is the corresponding scale space, and $\dot{\mathbf{u}}(t)$ is its derivative with respect to t . We denote the entries of an N -point signal by the same symbol as the signal itself, with additional subscripts 1 through N :

$$\begin{aligned} \mathbf{u}^0 &= (u_1^0, u_2^0, \dots, u_{N-1}^0, u_N^0)^T; \\ \mathbf{u}(t) &= (u_1(t), u_2(t), \dots, u_{N-1}(t), u_N(t))^T. \end{aligned}$$

Since most operators \mathcal{B}_1 of interest will involve first differences of the form $u_{n+1} - u_n$, it will simplify our notation to also define non-existent samples u_0 and u_{N+1} . Thus, all vectors will implicitly be $(N+2)$ -dimensional. Typically, we will take $u_0 = u_1$ and $u_{N+1} = u_N$. We emphasize that subscripts 0 through $N+1$ will always denote the samples of a signal, whereas the superscript 0 will be reserved exclusively to denote the signal which is the initial condition of a differential equation.

We similarly denote an N -by- N image to be processed by $\mathbf{u}^0 \in \mathbb{R}^{N^2}$; it will always be clear from the context whether \mathbf{u}^0 refers to a 1-D or a 2-D discrete signal. The corresponding system of ODEs is

$$\begin{aligned} \dot{\mathbf{u}}(t) &= \mathcal{B}_2(\mathbf{u}(t)) \\ \mathbf{u}(0) &= \mathbf{u}^0, \end{aligned} \quad (2.7)$$

where \mathbf{u}^0 and $\mathbf{u}(t)$ are matrices whose entries in the i -th row and j -th column are $u_{i,j}^0$ and $u_{i,j}(t)$, respectively.

The operators \mathcal{B}_1 and \mathcal{B}_2 will typically be the negative gradient of some energy functional, which we will denote by $\mathcal{E}(\mathbf{u})$. This energy will depend on the first differences of \mathbf{u} in the following way:

$$\mathcal{E}(\mathbf{u}) = \sum_{(s,r) \in \mathcal{N}} E(u_s - u_r), \quad (2.8)$$

where

- E is an even function;
- s and r are single indices if \mathbf{u} is a 1-D signal and pairs of indices if \mathbf{u} is a 2-D image;
- \mathcal{N} is the list of all neighboring pairs of pixels: s and r are neighbors if and only if $(s, r) \in \mathcal{N}$.

We will use the following neighborhood structure in 1-D:

$$\mathcal{N} = \{(n, n + 1)\}_{n=1}^{N-1}. \quad (2.9)$$

In other words, the sample at n has two neighbors: at $n - 1$ and at $n + 1$. We use a similar neighborhood structure in 2-D, where each pixel (i, j) has four neighbors: $(i - 1, j)$, $(i + 1, j)$, $(i, j - 1)$, and $(i, j + 1)$.

■ 2.2 Linear and Non-linear Diffusions.

To understand the conceptual basis of SIDs, it is useful to briefly review one of the lines of thought that has spurred work in evolution-based methods for image analysis. In [72] Witkin proposed filtering an original image $u^0(x, y)$ with Gaussian kernels of variance $2t$, to result in a one-parameter family of images $u(t, x, y)$ he referred to as “a scale space”. This filtering technique has both a very important interpretation and a number of significant limitations that inspired the search for alternative scale spaces, better adapted to edge detection and image segmentation. In particular, a major limitation is that linear Gaussian smoothing blurs and displaces edges, merges boundaries of objects that are close to each other, and removes edge junctions [55], as illustrated in Figure 2.1.

However, the important insight found, for example, in [35], is that the family of images $u(t, x, y)$ is the solution of the linear heat equation with $u^0(x, y)$ as the initial data:

$$\begin{aligned} u_t &= u_{xx} + u_{yy} \\ u(0, x, y) &= u^0(x, y), \end{aligned} \quad (2.10)$$

where the subscripts denote partial derivatives. This insight led to the pursuit and development of a new paradigm for processing images via the evolution of nonlinear

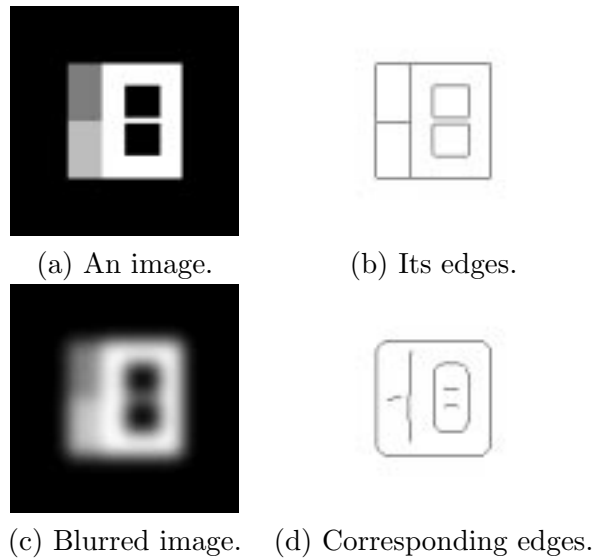


Figure 2.1. (a) An artificial image; (b) the edges corresponding to the image in (a); (c) the image in (a) blurred with a Gaussian kernel; (d) the edges corresponding to the blurred image. Note that T-junctions are removed, corners are rounded, and two black squares are merged together. The edges here are the maxima of the absolute value of the gradient.

PDEs [1, 46, 49, 50, 56] which effectively lift the limitations of the linear heat equation. For example, in [49, 50], Perona and Malik propose to achieve both noise removal and edge enhancement through the use of a non-uniform diffusion which in essence acts as an unstable inverse diffusion near edges and as a stable linear-heat-equation-like diffusion in homogeneous regions without edges:

$$\begin{aligned} u_t &= \vec{\nabla} \cdot \{G(|\nabla u|)\nabla u\}, \\ u(0, x, y) &= u^0(x, y), \end{aligned} \quad (2.11)$$

where $\vec{\nabla}$ and ∇ are the divergence (2.5) and gradient (2.3), respectively. The nonlinear

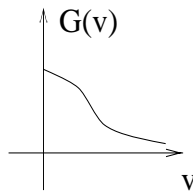


Figure 2.2. The G function from the right-hand side of the Perona-Malik equation (2.11).

diffusion coefficient $G(|\nabla u|)$ is chosen so as to suppress diffusion in the regions of high gradient which are identified with edges, and to encourage diffusion in low-gradient regions identified with noise (Figure 2.2). More formally, G is a nonnegative monoton-

ically decreasing function with $G(0) = 1$. (Note that if G were identically equal to 1, then (2.11) would turn into the linear heat equation (2.10), since $\vec{\nabla} \cdot (\nabla u) = u_{xx} + u_{yy}$.)

To simplify the analysis of the behavior of this equation near edges, we re-write it below in one spatial dimension; however, the statements we make also apply to 2-D.

$$\begin{aligned} u_t &= \frac{\partial}{\partial x} \{F(u_x)\} \\ u(0, x) &= u^0(x), \end{aligned} \quad (2.12)$$

where $F(u_x) = G(|u_x|)u_x$, i.e., F is odd, and tends to zero at infinity. Perona and Malik also impose that F have a unique maximum at some location K (Figure 2.3). This constant K is the threshold between diffusion and enhancement, in the following sense. If, for a particular time $t = t_0$, we define an “edge” of $u(t_0, x)$ as an inflection



Figure 2.3. The F function from the right-hand side of the Perona-Malik equation (2.12).

point with the property $u_x u_{xxx} < 0$, then a simple calculation shows that all such edges where $|u_x| < K$ will be diminished by (2.12)—i.e. $|u_x|$ will be reduced, while the larger edges, with $|u_x| > K$, will be enhanced. It has been observed [34] that the numerical implementations of (2.12) do not exactly exhibit this behavior, although they do produce temporary enhancement of edges, resulting in both noise removal and scale spaces in which the edges are much more stable across scale than in linear scale spaces. As Weickert pointed out in [69], “a scale-space representation cannot perform better than its discrete realization”. These observations naturally led to a closer analysis (described in the next chapter) of a semi-discrete counterpart of (2.12), i.e., of the following system of ordinary differential equations:

$$\begin{aligned} \dot{u}_n &= F(u_{n+1} - u_n) - F(u_n - u_{n-1}), \quad n = 1, \dots, N, \\ \mathbf{u}(0) &= \mathbf{u}^0, \end{aligned} \quad (2.13)$$

where $\mathbf{u}^0 = (u_1^0, \dots, u_N^0)^T \in \mathbb{R}^N$ is the signal to be processed, and where the conventions $u_{N+1} = u_N$ and $u_0 = u_1$ are used.

The 2-D semi-discrete version of the Perona-Malik equation is similar:

$$\begin{aligned} \dot{u}_{ij} &= F(u_{i+1,j} - u_{ij}) - F(u_{ij} - u_{i-1,j}) \\ &\quad + F(u_{i,j+1} - u_{ij}) - F(u_{ij} - u_{i,j-1}), \\ \mathbf{u}(0) &= \mathbf{u}^0, \end{aligned}$$

with $i = 1, 2, \dots, N$, $j = 1, 2, \dots, N$, and with the conventions $u_{0,j} = u_{1,j}$, $u_{N+1,j} = u_{N,j}$, $u_{i,0} = u_{i,1}$ and $u_{i,N+1} = u_{i,N}$.

■ 2.3 Region Merging Segmentation Algorithms.

It is shown in the next chapter that SIDEs can be classified under a very broad category of multiscale region merging segmentation methods.

The goal of image segmentation is to partition the domain Ω of definition of a given image \mathbf{u}^0 into several disjoint regions O_1, \dots, O_k ($\bigcup_{i=1}^k O_i = \Omega$), such that \mathbf{u}^0 is, in some sense, homogeneous within each region. Many segmentation algorithms also provide a filtered version \mathbf{u} of \mathbf{u}^0 , which is in essence a smoothing of \mathbf{u}^0 within each region and has abrupt changes between regions. A boundary between two such neighboring regions is called *an edge*. If a segmentation E_2 can be obtained from a segmentation E_1 by erasing edges, it is said that E_2 is *coarser* than E_1 . A *multiscale segmentation algorithm* provides a hierarchy of segmentations which get progressively coarser. Morel and Solimini's [43] definition of a generic multiscale region merging algorithm for image segmentation is paraphrased and augmented below.

1. Initialize the algorithm with the finest possible segmentation (i.e., each pixel is a separate region). Fix the scale t at a very small value t_0 .
2. Merge all pairs of regions whose merging "improves" the segmentation. If the coarsest segmentation is obtained (i.e., the whole domain Ω is the one and only region), stop.
3. If necessary, update the estimate \mathbf{u} .
4. Increment the scale parameter t .
5. Go to Step 2.

Numerous algorithms have been constructed that fall under this, very general, paradigm. Their principal points of difference are Step 3 and the merging criterion used in Step 2. Two instances of such algorithms are Pavlidis' algorithm of 1972 [47] and Koepfler, Lopez, and Morel's algorithm of 1994 [36].

Example 2.1. *Pavlidis' algorithm.*

Step 2 of Pavlidis' algorithm consists of merging two neighboring regions, O_i and O_j , if the variance of \mathbf{u}^0 over $O_i \cup O_j$ is less than t . Step 3 is omitted. ■

Example 2.2. *The algorithm of Koepfler, Lopez, and Morel.*

Koepfler, Lopez, and Morel use the piecewise-constant Mumford-Shah model [44] as their merging criterion. They form the filtered image \mathbf{u} by assigning to every pixel in a region O_i the average value of \mathbf{u}^0 over that region. Two regions, O_i and O_j , with respective average values u_i and u_j , are merged by removing the boundary between them and replacing both u_i and u_j with their weighted average, $(|O_i|u_i + |O_j|u_j)/(|O_i| + |O_j|)$, provided that the global energy is reduced:

$$(\mathbf{u} - \mathbf{u}^0)^T(\mathbf{u} - \mathbf{u}^0) + tl.$$

Here, $|O_p|$ is the number of pixels in the region O_p and l is the total length of all the edges.

This method admits fast numerical implementations and has been experimentally shown to be robust to white Gaussian noise. However, as we will illustrate, the quadratic penalty on the disagreement between the estimate \mathbf{u} and the initial data \mathbf{u}^0 renders it ineffective against more severe noise, such as speckle encountered in SAR images. ■

Note that region merging methods do not allow edges to be created. Thus, decisions made in the beginning of an algorithm cannot be undone later. A slight modification of such methods results in split-and-merge methods, which combine region growing with region splitting [43].

■ 2.4 Shock Filters and Total Variation.

We will see in the next chapter that SIDEs blend together elements of the total variation minimization [6, 56, 58] and shock filters [46]. A shock filter is a PDE which was introduced by Osher and Rudin in [46] to achieve signal and image enhancement (de-blurring). Since its 2-D version will not be needed in this thesis, we restrict our discussion to one spatial dimension:

$$\begin{aligned} u_t &= -|u_x|f(u_{xx}), \\ u(0, x) &= u^0(x), \end{aligned} \tag{2.14}$$

where $f(0) = 0$, and the function f has the same sign as its argument: $\text{sgn}(f(u_{xx})) = \text{sgn}(u_{xx})$ for $u_{xx} \neq 0$. This PDE has the property of developing shocks (jumps) near the points of inflection of the initial condition. The most serious limitation of the shock filters, acknowledged in [46], is non-robustness to noise. In fact, they enhance noise, together with the “useful” edges. There are, however, stable numerical schemes, which—similarly to the semi-discrete Perona-Malik equation—do not exhibit certain properties of the underlying PDE. In particular, for certain choices of f , the scheme presented in [46] sharpens noise-free signals, but leaves noisy signals virtually unchanged. This is because every local extremum remains stationary under this scheme. This behavior is illustrated in Figure 2.4, which presents the results of the simulations for the equation

$$u_t = -|u_x|\text{sgn}(u_{xx}). \tag{2.15}$$

For other choices of f , the numerical scheme of [46], in addition to keeping the local extrema stationary, creates other local extrema and is unstable, which is illustrated in Figure 2.5 for the equation

$$u_t = -|u_x|u_{xx}. \tag{2.16}$$

Total variation minimization is another restoration technique, developed by the same authors in [56] and independently by Bouman and Sauer in [6, 58] in a different

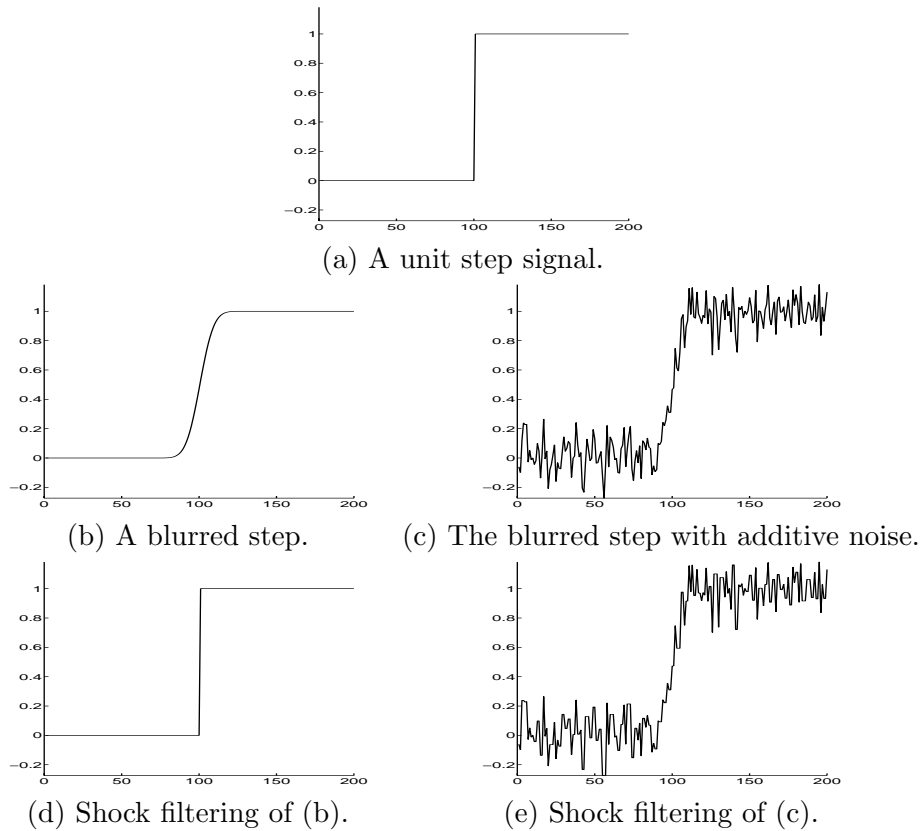
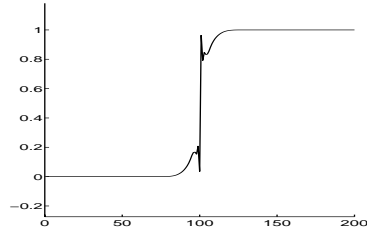


Figure 2.4. (b) Gaussian blurring of the signal depicted in (a). (c) Signal depicted in (b), with additive white Gaussian noise of variance 0.1. (d) The steady state of the shock filter (2.15), with the signal (b) as the initial condition. The reconstruction is perfect, modulo numerical errors. (e) The steady state of the shock filter (2.15), with the signal (c) as the initial condition. It is virtually the same as (c), since all extrema remain stationary.

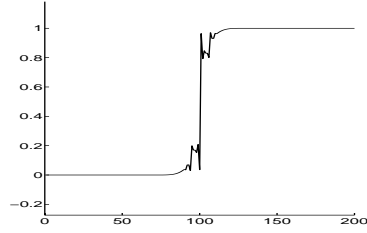
context and in a somewhat different form. We start with Bouman and Sauer’s work, since it was chronologically first, and since—as we will see in the next chapter—it is conceptually closer to the results presented in this thesis.

The objective of [6, 58] is reconstructing an image \mathbf{u} from its tomographic projections \mathbf{u}^0 . The authors consider transmission tomography, where the projection data are in the form of the number of photons detected after passing through an absorptive material. In other words, \mathbf{u}^0 is the number of photon counts for each angle and displacement. Bouman and Sauer use a probabilistic setting, where the photon counts are Poisson random variables, independent among angles and displacements. They derive an expression for the log likelihood function $L(\mathbf{u}|\mathbf{u}^0)$, and seek the maximum *a posteriori* [66] estimate $\hat{\mathbf{u}}$ of \mathbf{u} :

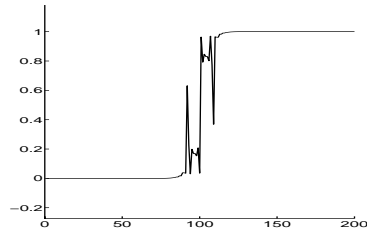
$$\hat{\mathbf{u}} = \arg \max_{\mathbf{u}} \{L(\mathbf{u}|\mathbf{u}^0) + \mathcal{E}(\mathbf{u})\}, \quad (2.17)$$



(a) Five iterations of the shock filter (2.16).



(a) Ten iterations of the shock filter (2.16).



(a) Eighteen iterations of the shock filter (2.16).

Figure 2.5. Filtering the blurred unit step signal of Figure 2.4, (b) with the shock filter (2.16): (a) 5 iterations, (b) 10 iterations, (c) 18 iterations. Spurious maxima and minima are created; the unit step is never restored.

where $\mathcal{E}(\mathbf{u})$ is the logarithm of the prior density function of \mathbf{u} . They propose the following prior model:

$$\mathcal{E}(\mathbf{u}) = \gamma \sum_{(s,r) \in \mathcal{N}} |u_s - u_r|, \quad (2.18)$$

where \mathcal{N} is the list of all neighboring pairs of pixels, and γ is a constant.

The optimization problem (2.17) cannot be solved by gradient descent, since $\mathcal{E}(\mathbf{u})$ is not differentiable at the points where $u_s = u_r$ for $(s, r) \in \mathcal{N}$. Bouman and Sauer improve upon the existing iterative methods for solving non-differentiable optimization problems [4, 54, 73] by introducing a technique which they call *segmentation based optimization*. The basic idea is to combine a Gauss-Seidel type approach [59] with a split-and-merge segmentation strategy. Specifically, if the image values at two neighboring locations are both equal to some number α , then α is changed until a minimum of the objective function (2.17) is achieved. Similarly, if several neighboring pixels $u_{s_1} = u_{s_2} = \dots = u_{s_i}$ have the same value, they are grouped together and changed together by moving

along the hyperplane $\{\mathbf{u} : u_{s_1} = u_{s_2} = \dots = u_{s_i}\}$ until a minimum of the objective function (2.17) is achieved. After each pixel of the image is visited in such a manner, a “split” iteration follows, where each pixel is freed to seek its own conditionally optimal value. This approach is theoretically justified and extended in Chapter 3, where it is shown that the steepest descent for a non-differentiable energy function such as (2.18) is a differential equation which automatically merges pixels, thereby segmenting the underlying image.

We also point out that the continuous version of the energy (2.18) is

$$\int |u_x| dx \quad \text{in 1-D,}$$

$$\text{and } \int \int |\nabla u| dx dy \quad \text{in 2-D,}$$

and is called *the total variation of u*. Its constrained minimization was used in [56] for image restoration. The restored version $u(x, y)$ of an image $u^0(x, y)$ was computed by solving the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & \int \int |\nabla u| dx dy & (2.19) \\ \text{subject to} \quad & \int \int (u - u^0) dx dy = 0 \\ \text{and} \quad & \int \int (u - u^0)^2 dx dy = \sigma^2. \end{aligned}$$

■ 2.5 Constrained Restoration of Geman and Reynolds.

We will see in the next chapter that a 1-D SIDE is the gradient descent equation for the global energy $\mathcal{E}(\mathbf{u}) = \sum E(u_{i+1} - u_i)$, where $E(v)$ is concave everywhere except at zero and non-differentiable at zero, and looks like Υ (for example, $E(v) = \arctan(v)$ or $E(v) = 1 - (1 + |v|)^{-1}$ —see Figure 2.6). This energy is similar to the first term of the image restoration model of D. Geman and Reynolds [21]. It is also interesting to note that the potential function of the Gibbs distribution learned from natural images in [75] has the same basic Υ -shape. This indicates that the functionals involving such a term may be the right ones for modeling natural images.

■ 2.6 Conclusion.

An exhaustive survey of variational models in image processing is beyond the scope of this thesis. A much more complete bibliography can be found in [43]. In particular, Chapter 3 of [43] contains a very nice discussion of region merging segmentation algorithms, starting with Brice and Fennema’s [7] and Pavlidis’ [47], which may be considered as ancestors to both [36], snakes [31], and SIDEs. Examples of more recent algorithms, not covered in [43], are [16] and [24]. Another important survey text,

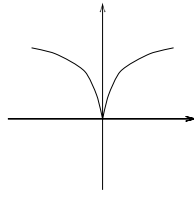


Figure 2.6. The SIDE energy function, also encountered in the models of Geman and Reynolds, and Zhu and Mumford.

which also contains a wealth of references both on variational methods and nonlinear diffusions, is [55].