CHAPTER 2

EARLY VISUAL PROCESSING

In the previous chapter, the traditional bottom-up paradigm that dominated the studies in computational vision was analyzed, and some interesting and challenging cases for the bottom-up segmentation approaches were presented that motivate the proposed works in this thesis.

These challenges generally stem from the unidirectional information flow. For instance, consider detecting edges that plays a key role in image segmentation. Since edges are described as abrupt intensity changes in the images, edge detection in fact requires a kind of derivative computation through local edge operators. If the image data is noisy, differentiation yields inaccurate results. Hence, the input image needs to be smoothed out before applying the operators. As discussed in the previous chapter, this leads to a *chicken and egg dilemma* since the smoothing process should eliminate the noise, and yet preserve the edges. Needless to say, this requires a clear understanding of possible edge points which is the actual goal of edge detection.

About twenty years ago, the dilemma in the one way information flow between smoothing and edge detection prompted eminent researchers to propose methods that combine smoothing and edge detection [20, 62, 100, 109]. Most of these unified frameworks are based on nonlinear diffusion models and variational regularization/segmentation methods. Although, partial differential equations (PDEs), which appear in many diffusion models, and variational approaches have been widely and successfully used in areas such as physics and engineering for many years, they have been an area of intensive research in image processing and computer vision only for the past two decades, yet they have provided effective solutions especially for image smoothing and restoration tasks.

We prefer to start our review with the linear diffusion equation to provide a motivation for these studies. It is also used as an illustrative example to analyze the relation between

variational regularization frameworks and diffusion equations. The subsequent parts of this chapter are devoted to important nonlinear diffusion models and variational regularization methods. The numerical implementation details of some of these works are also provided since they are closely related with the proposed works. The chapter concludes with a brief discussion on the reviewed works.

2.1 Linear Diffusion

The linear diffusion (heat) equation is the oldest and best investigated PDE method in image processing. Let $f(x)$ denote a grayscale (noisy) input image and $u(x,t)$ be initialized with $u(x,0) = u^0(x) = f(x)$. Then, the linear diffusion process can be defined by the equation

$$\frac{\partial u}{\partial t} = \nabla \cdot (\nabla u) = \nabla^2 u \tag{2.1}$$

where $\nabla\cdot$ denotes the divergence operator. Hence,

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \, . \tag{2.2}$$

The diffusion process can be seen as an evolution process with an artificial time variable t denoting the *diffusion time* where the input image is smoothed at a constant rate in all directions. Starting from the initial image $u^0(x)$, the evolving images $u(x,t)$ under the governed equation represent the successively smoothed versions of the initial input image $f(x)$, and thus create a *scale space* representation of the given image f, with $t > 0$ being the scale. As we move to coarser scales, the evolving images become more and more simplified since the diffusion process removes the image structures at finer scales. Figure 2.1 and Figure 2.2 show example scale space representations sampled at different diffusion times for two different images. In fact, the notion of scale is an essential part of early visual processing, where the main task is to separate the image into relevant and irrelevant parts.

It is shown that the solution of the linear diffusion equation with the given initial condition $u(x,0) = f(x)$ for a specific diffusion time T is equivalent to the convolution of the input image $f(x)$ with the Gaussian kernel $G_\sigma(x)$ with standard deviation $\sigma = \sqrt{2T}$ [74, 79, 155]. Thus, linear diffusion can be regarded as a low-pass filter. The correspondence between the diffusion time variable t and the standard deviation σ clearly depicts the effect of t on the evolving images. The higher the value of t, the higher the value of σ, and the more smooth the image becomes. This relation also provides the following explicit solution to (2.1):

$$u(x,t) = \left(G_{\sqrt{2T}} * g\right)(x) \qquad \text{with} \qquad G_\sigma(x) = \frac{1}{2\pi\sigma^2} exp\left(-\frac{|x|^2}{2\sigma^2}\right). \tag{2.3}$$

Figure 2.1: Linear diffusion results for different diffusion times.

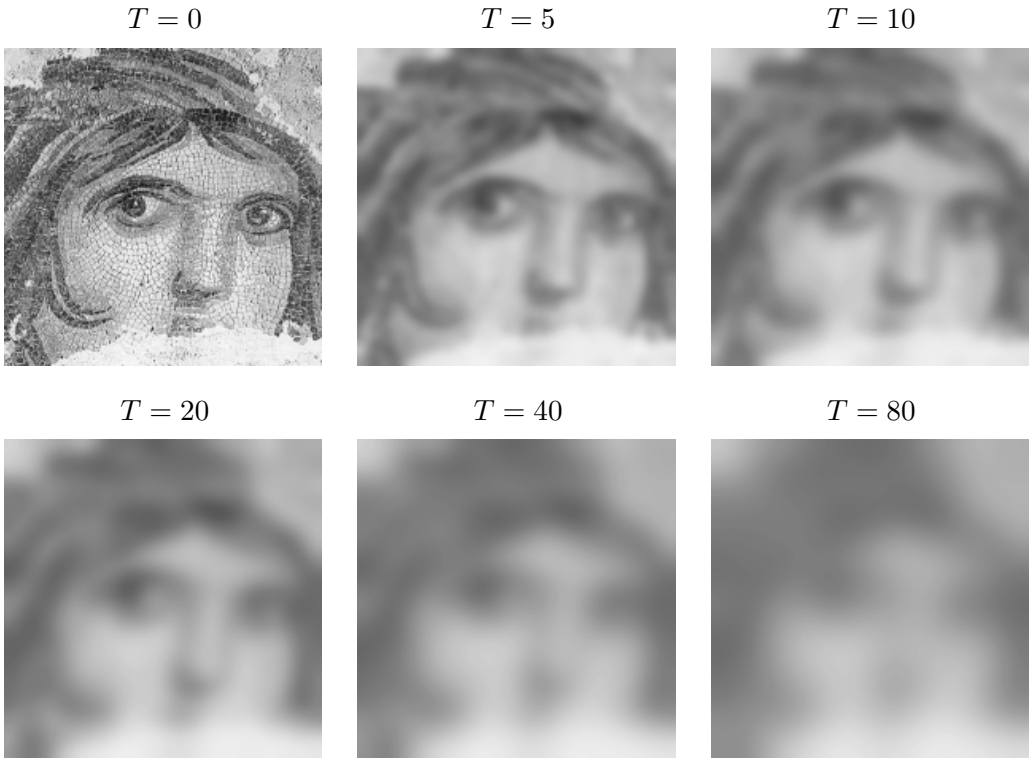


Figure 2.2: Linear diffusion results for different diffusion times.

Numerical Implementation

Since we deal with digital images, solving the linear diffusion equation requires discretization in both spatial and time coordinates. Central differences are the typical choices for the spatial derivatives:

$$\frac{d^2u_{i,j}}{dx^2} \approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_x^2}, \qquad \frac{d^2u_{i,j}}{dy^2} \approx \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h_y^2} \tag{2.4}$$

where $u_{i,j}$ denotes the gray value or the brightness of the evolving image at pixel location (i, j).

The values of h_x and h_y are generally set to 1 as digital images are discretized on a regular pixel grid. For the remainder of this thesis, we take $h_x = h_y = 1$. This leads to the following space-discrete equation for (2.1):

$$\frac{du_{i,j}}{dt} = u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j} \ . \tag{2.5}$$

The straightforward approach to solve (2.5) is to consider an iterative scheme with an explicit time discretization, where homogeneous Neumann boundary condition is imposed along the image boundary

$$\frac{u_{i,j}^{k+1} - u_{i,j}^{k}}{\Delta t} = u_{i+1,j}^{k} + u_{i-1,j}^{k} + u_{i,j+1}^{k} + u_{i,j-1}^{k} - 4u_{i,j}^{k} \tag{2.6}$$

where Δt is the time step, and u^k represents the restored image u at iteration k. Numerical stability condition for the discrete scheme requires that $\Delta t \leq 0.25$.

Relation Between Variational Regularization and Diffusion Equations

Interestingly, there is a strong relation between variational regularization methods and diffusion equations [124]. The variational regularization methods formulate smoothing process as a functional minimization via which a noise-free approximation of a given image is to be estimated. Most of these formulations assume an additive noise model

$$f(x) = u(x) + n(x) \tag{2.7}$$

where $f(x)$ and $u(x)$ respectively denote the given noisy image and the desired denoised image, and $n(x)$ represents the additive noise.

Consider the Tikhonov energy functional [145] as an illustrative example:

$$E(u) = \int_{\Omega} \left((u - f)^2 + \alpha |\nabla u|^2 \right) dx \tag{2.8}$$

where

- $\Omega \subset \mathbb{R}^2$ is connected, bounded, open subset representing the image domain,
- f is an image defined on Ω,
- u is the smooth approximation of f,
- $\alpha > 0$ is the scale parameter.

The first term in $E(u)$ is the *data fidelity* term that penalizes the deviations between u and f, and thus forces the restored image to be close to the original image. The second term is called the *regularization* or *smoothness* term which penalizes the high gradients, and gives preference to smooth approximations. The relative importance of these two terms are defined by the scale parameter α.

The minimizing function u formally satisfies the Euler-Lagrange equation

$$(u - f) - \alpha \nabla^2 u = 0 \tag{2.9}$$

with the Neumann boundary condition $\frac{\partial u}{\partial n}\big|_{\partial\Omega} = 0$.

It is possible to rewrite (2.9) as

$$\frac{u - u^0}{\alpha} = \nabla^2 u \qquad \text{with} \qquad u^0 = f, \tag{2.10}$$

which may be regarded as an implicit time discretization of the linear diffusion equation (2.1) where a single time step ($T = \alpha$) is used. Note that diffusion time (scale selection) problem is not really eliminated by the variational regularization, it is replaced with a new parameter α that determines the strength of the smoothness prior.

The main drawback of linear diffusion filtering is that the smoothing process does not consider information regarding important image features such as edges. It follows that same amount of smoothing to be applied at every image location. As a result, the diffusion process does smooth not only noise, but also image edges.

2.2 Perona-Malik Type Nonlinear Diffusion [109]

The main theory behind nonlinear diffusion models is to use nonlinear PDEs to create a scale space representation that consists of gradually simplified images where some image features such as edges are maintained or even enhanced. The earliest nonlinear diffusion model proposed in image processing is the so-called *anisotropic diffusion*[1] by Perona and

[1] In fact, Perona-Malik equation is an isotropic nonhomogeneous equation as it uses a scalar-valued diffusivity. A true example of anisotropic diffusion model, edge-enhancing diffusion [153], will be summarized in Section 2.6.

Malik [109].

In their formulation, they replaced the constant diffusion coefficient of linear equation (2.1) by a smooth nonincreasing diffusivity function g with $g(0) = 1$, $g(s) \geq 0$, and $\lim_{s\to\infty} g(s) = 0$. As a consequence, the diffusivities become variable in both space and time. The Perona-Malik equation is

$$\frac{\partial u}{\partial t} = \nabla \cdot (g(|\nabla u|)\nabla u) \tag{2.11}$$

with homogeneous Neumann boundary conditions and the initial condition $u^0(x) = f(x)$, f denoting the input image.

Perona and Malik suggested two different choices for the diffusivity function:

$$g(s) = \frac{1}{1 + s^2/\lambda^2}\,, \tag{2.12}$$

$$g(s) = e^{-\frac{s^2}{\lambda^2}} \tag{2.13}$$

where λ corresponds to a contrast parameter. These functions share similar characteristics, and result in similar effects on the diffusivities.

We review the 1D physical analysis of the Perona-Malik diffusion below since it clearly demonstrates the role of the contrast parameter λ and the main behavior of the equation [154]. For 1D case, the Perona-Malik equation is as follows:

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \underbrace{(g(|u_x|)u_x)}_{\Phi(u_x)} = \Phi'(u_x)u_{xx} \tag{2.14}$$

with $g(|u_x|) = \frac{1}{1+|u_x|^2/\lambda^2}$ or $g(|u_x|) = e^{-\frac{|u_x|^2}{\lambda^2}}$.

Figure 2.3 shows the diffusivity functions and the corresponding flux functions for linear diffusion and Perona-Malik type nonlinear diffusion. One can easily observe that for linear diffusion the diffusivity is constant ($g(s) = 1$), which results in a linearly increasing flux function. As a result, all points, including the discontinuities, are smoothed equally. For Perona-Malik, the diffusivity is variable and decreases as $|u_x|$ increases. It is evident that the decay in diffusivity is particularly rapid after the contrast parameter λ. This leads to two different behaviors in the diffusion process. Since $\frac{\partial u}{\partial t} = \Phi'(u_x)u_{xx}$, for the points where $|u_x| < \lambda$, $\Phi'(u_x) > 0$ which corresponds to lost in the material. For the points where $|u_x| > \lambda$, on the contrary, $\Phi'(u_x) < 0$ which generates an enhancement in the material. Hence, although the diffusivity is always nonnegative, one can observe both *forward* and *backward* diffusions during the smoothing process, and the contrast parameter λ separates the regions of forward diffusion from the regions of backward diffusion.

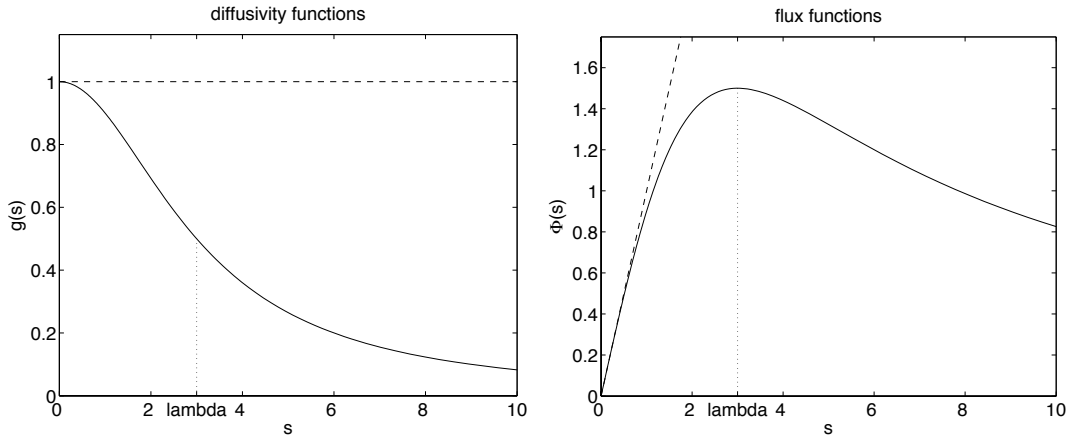


Figure 2.3: Diffusivities and the corresponding flux functions for linear diffusion (*plotted in dashed line*) and Perona-Malik type nonlinear diffusion (*plotted in solid line*). For Perona-Malik diffusivity $g(s) = \frac{1}{1+s^2/\lambda^2}$ is used with $\lambda = 3$.

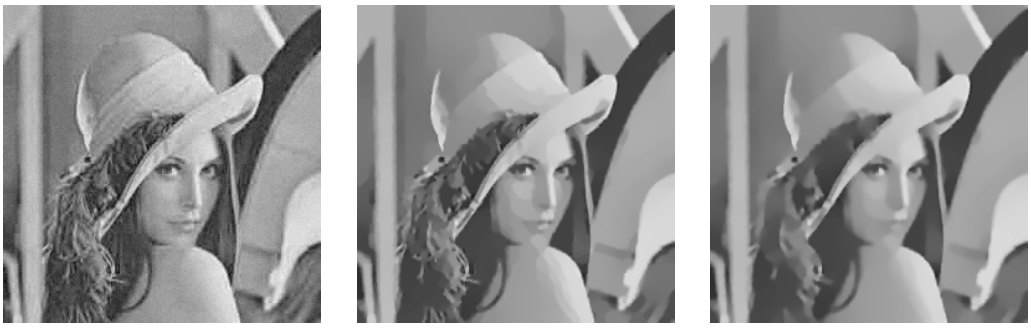
If we consider the 2D case, the diffusivities are reduced at the image locations where $|\nabla u|^2$ is large. As $|\nabla u|^2$ can be interpreted as a measure of edge likelihood, this means that the amount of smoothing is low along image edges. In particular, the contrast parameter λ specifies a measure that determines which edge points are to be preserved or blurred during the diffusion process. Even edges can be sharpened due to the local backward diffusion behavior as discussed for the 1D case. Since the backward diffusion is a well-known ill-posed process, this may cause an instability, the so-called *staircasing effect*, where a piece-wise smooth region in the original image evolves into many unintuitive piecewise constant regions. Figure 2.4 shows an example where this instability occurs. The unintuitive regions such as the one at the woman's face and shoulder are clearly visible in Figure 2.4(b). A possible solution to this drawback is to use regularized gradients in diffusivity computations [36] (Figure 2.4(c)).

Replacing the diffusivities $g(|\nabla u|)$ with the regularized ones $g(|\nabla u_\sigma|)$ leads to the following equation:

$$\frac{\partial u}{\partial t} = \nabla \cdot (g(|\nabla u_\sigma|)\nabla u) \tag{2.15}$$

where $u_\sigma = G_\sigma * u$ represents a Gaussian-smoothed version of the image. Taking the equivalence of the Gaussian smoothing and the linear scale space into account, ∇u_σ can also be considered as the gradient computed at a specific scale $\sigma > 0$.

Some example results of regularized Perona-Malik filtering with different diffusion times are shown in Figure 2.5 and Figure 2.6. It is evident from these images that the corresponding smoothing process diminishes noise while retaining or even enhancing edges since it considers



(a) (b) (c)

Figure 2.4: The staircasing effect. (a) Original noisy image. (b) Perona-Malik diffusion. (c) Regularized Perona-Malik diffusion.

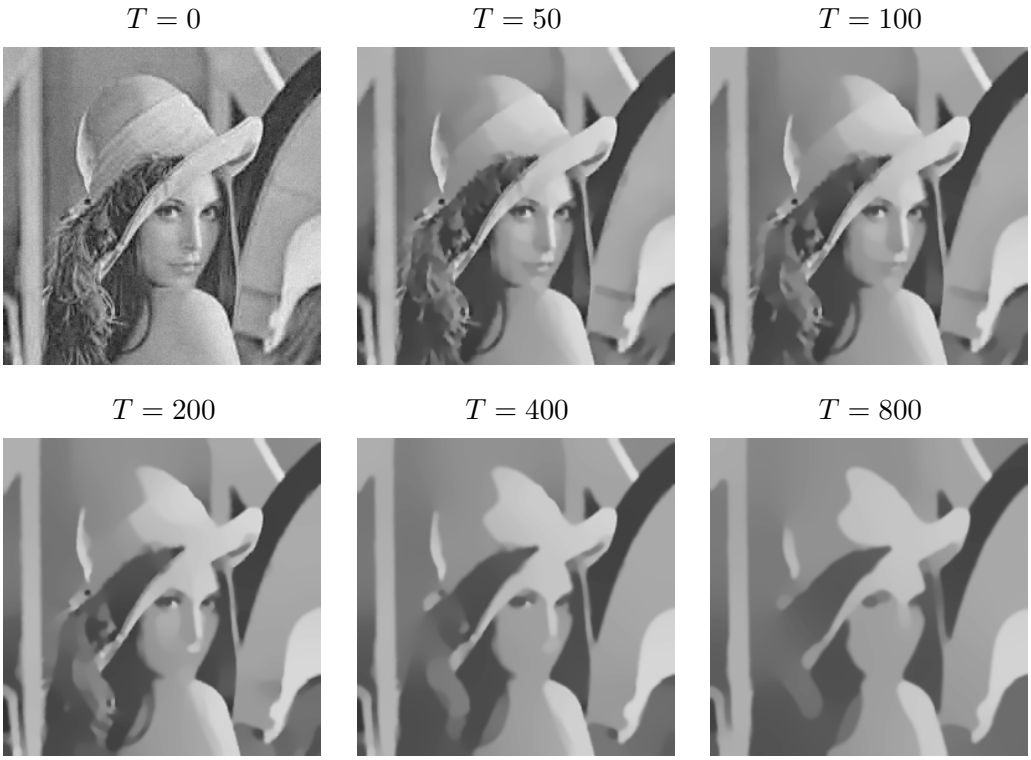


Figure 2.5: Regularized Perona-Malik results for different diffusion times ($\lambda = 1$, $\sigma = 1$).

a kind of a priori edge knowledge.

Numerical Implementation

For numerical implementation, we use central differences to approximate the gradient magnitude at a pixel (i, j) in the diffusivity estimation, $g_{i,j} = g(|\nabla u_{i,j}|)$:

$$|\nabla u_{i,j}| = \sqrt{\left(\frac{du_{i,j}}{dx}\right)^2 + \left(\frac{du_{i,j}}{dy}\right)^2} \approx \sqrt{\left(\frac{u_{i+1,j} - u_{i-1,j}}{2}\right)^2 + \left(\frac{u_{i,j+1} - u_{i,j-1}}{2}\right)^2}. \quad (2.16)$$

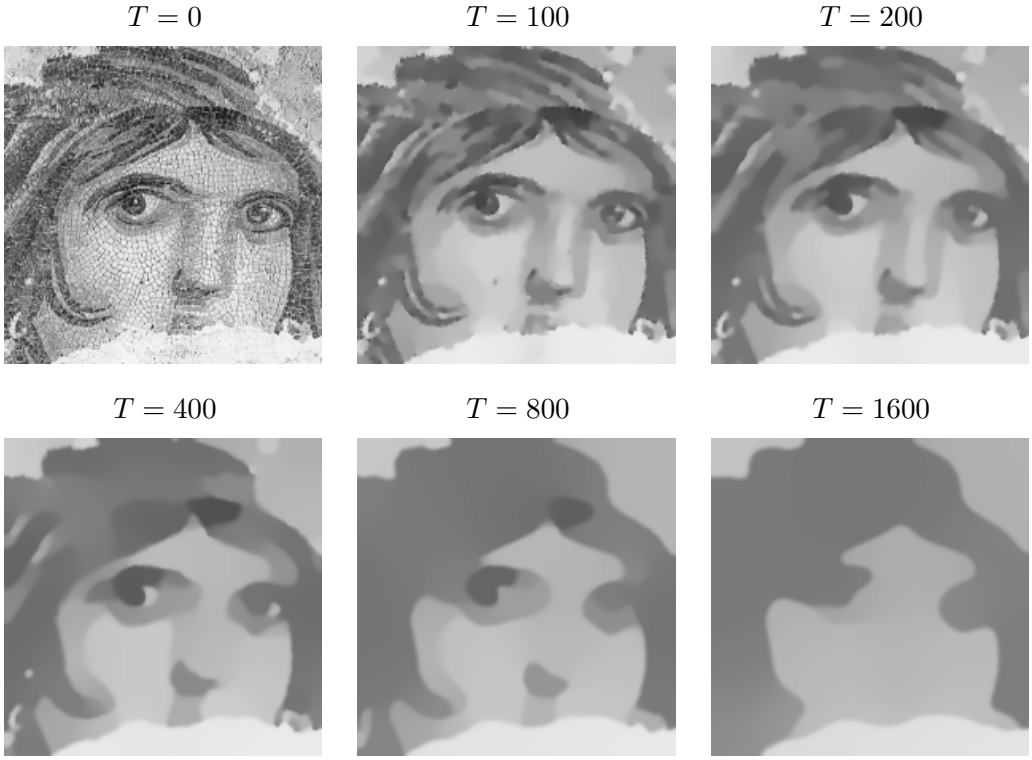


Figure 2.6: Regularized Perona-Malik results for different diffusion times ($\lambda = 1$, $\sigma = 1$).

The Perona-Malik equation (2.11) is first discretized w.r.t. spatial variables. This results in the following space-discrete equation:

$$\begin{aligned} \frac{\partial u}{\partial t} &= \frac{\partial}{\partial x}\left(g(|\nabla u|)u_x\right) + \frac{\partial}{\partial y}\left(g(|\nabla u|)u_y\right), \\ \frac{du_{i,j}}{dt} &= g_{i+\frac{1}{2},j}\cdot(u_{i+1,j}-u_{i,j}) - g_{i-\frac{1}{2},j}\cdot(u_{i,j}-u_{i-1,j}) \\ &+ g_{i,j+\frac{1}{2}}\cdot(u_{i,j+1}-u_{i,j}) - g_{i,j-\frac{1}{2}}\cdot(u_{i,j}-u_{i,j-1}). \end{aligned} \tag{2.17}$$

This discretization scheme requires the diffusivities to be estimated at mid-pixel points (Figure 2.7). They are simply computed by taking averages of the diffusivities over neighboring pixels:

$$g_{i\pm\frac{1}{2},j} = \frac{g_{i\pm1,j}+g_{i,j}}{2}, \qquad g_{i,j\pm\frac{1}{2}} = \frac{g_{i,j\pm1}+g_{i,j}}{2}. \tag{2.18}$$

The time derivative in (2.17) can be discretized using forward difference. This yields an iterative scheme with an explicit time discretization, where homogeneous Neumann boundary condition is imposed along the image boundary

$$\begin{aligned} \frac{u_{i,j}^{k+1}-u_{i,j}^{k}}{\Delta t} &= g_{i+\frac{1}{2},j}^{k}\cdot u_{i+1,j}^{k} + g_{i-\frac{1}{2},j}^{k}\cdot u_{i-1,j}^{k} + g_{i,j+\frac{1}{2}}^{k}\cdot u_{i,j+1}^{k} + g_{i,j-\frac{1}{2}}^{k}\cdot u_{i,j-1}^{k} \\ &- \left(g_{i+\frac{1}{2},j}^{k} + g_{i-\frac{1}{2},j}^{k} + g_{i,j+\frac{1}{2}}^{k} + g_{i,j-\frac{1}{2}}^{k}\right)\cdot u_{i,j}^{k} \end{aligned} \tag{2.19}$$

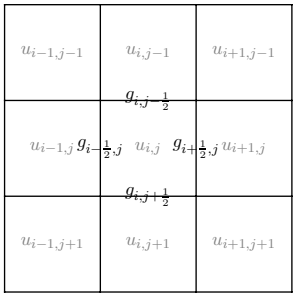


Figure 2.7: Discretization grid used in (2.17).

with Δt denoting the time step. For the Perona-Malik diffusion, the stability requirement is again $\Delta t \leq 0.25$.

2.3 Mumford-Shah (MS) Functional [100]

The formulation of Mumford and Shah [100] is based on a functional minimization via which a piecewise smooth approximation of a given image and an edge set are to be recovered simultaneously. In this unified formulation, smoothing and edge detection processes work jointly to partition an image into segments. The Mumford-Shah (MS) model is:

$$E_{MS}(u,\Gamma) = \beta \int_{\Omega} (u-f)^2 dx + \alpha \int_{\Omega\backslash\Gamma} |\nabla u|^2 dx + length(\Gamma) \tag{2.20}$$

where

- $\Omega \subset \mathbb{R}^2$ is connected, bounded, open subset representing the image domain,
- f is an image defined on Ω,
- $\Gamma \subset \Omega$ is the edge set segmenting Ω,
- u is the piecewise smooth approximation of f,
- α, $\beta > 0$ are the scale space parameters of the model.

The first term in E_{MS} is the *data fidelity* term which forces u to be close to the original image f. The next two terms are the generic priors that provide certain knowledge about the solution. Specifically, the second term, the so-called *regularization* or *smoothness* term, gives preference to piecewise smooth images by penalizing high gradients. Since the integral is over $\Omega\backslash\Gamma$, this prior is turned off at image boundaries, and thus it excludes image edges to be smoothed out. The third term is a penalty term on total edge length which prevents