Smoothing Filters

Tilakarathna. U. A. 200664P

Why Smoothing?

To remove noise.

Content

Blurring

- Usually implemented as a convolution with a kernel.
- **Attenuates high frequencies.**

1. Discrete Gaussian

- The size of the kernel is extended until there are enough discrete points in the Gaussian to ensure that a user-provided maximum error is not exceeded.
- Different Orders are used and following property holds.

$$\frac{d}{dx}(f * g) = f * \frac{d}{dx} g$$

- Longer computational time when variance is high

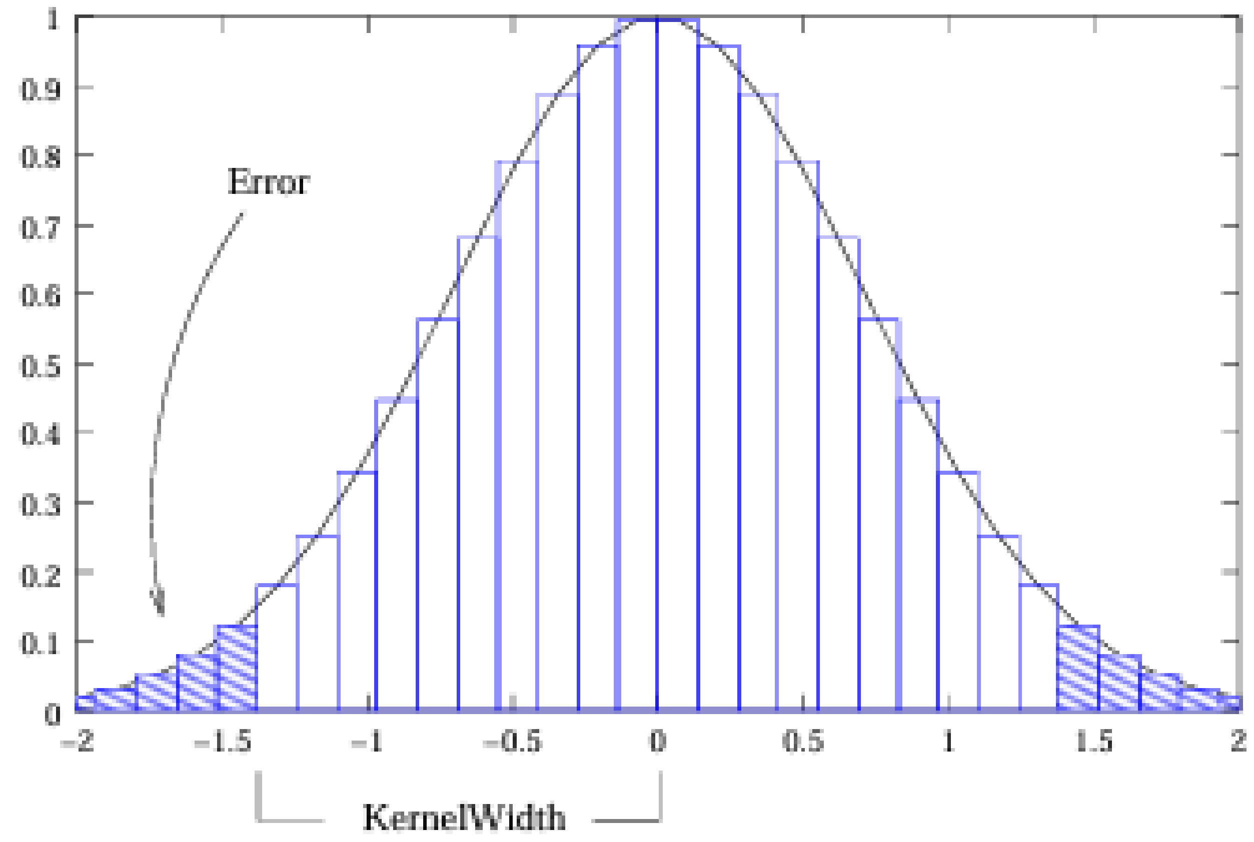

Figure 2.20: Discretized Gaussian.

2. Binomial Blurring

- Computes a nearest neighbor average along each dimension. The process is repeated a number of times, as specified by the user.

- Converges to Gaussian with the iterations.

3. Recursive Gaussian IIR

- Approximates convolution with Gaussian and its derivatives by IIR filters.

- Lesser computational time.

- Two normalization techniques to preserve maximum value or integral.

$$\frac{1}{\sigma\sqrt{2\pi}} \qquad \frac{1}{\sigma^2\sqrt{2\pi}}$$

Edge Preserving Smoothing

Anisotropic Diffusion

- Gaussian smoothed image is a single time slice of the solution to the heat equation, that has the original image as its initial conditions.
- Here, G(σ) is a Gaussian with standard deviation σ.

$$\frac{\partial g(x,y,t)}{\partial t} = \nabla \cdot \nabla g(x,y,t),$$

$g(x,y,0) = f(x,y)$ is the input image, is $g(x,y,t) = G(\sqrt{2t}) \otimes f(x,y)$.

Anisotropic Diffusion

- Anisotropic diffusion includes **a variable conductance term that, in turn, depends on the differential structure of the image.** It can be used to limit the smoothing at “edges” in images, as measured by high gradient magnitude.

$$g_t = \nabla \cdot c(|\nabla g|)\nabla g,$$

- The function c(|∇g|) is a fuzzy cutoff that reduces the conductance at areas of large |∇g|, and can be any one of a number of functions.

$$c(|\nabla g|) = e^{-\frac{|\nabla g|^2}{2k^2}}$$

Conductance Parameter: k
Time parameter: t

Anisotropic Diffusion

- These nonlinear partial differential equations can be solved on a discrete grid using finite forward differences. Thus, the smoothed image is obtained only by an **iterative process**, not a convolution or non-stationary, linear filter.

1. Gradient Anisotropic Diffusion

- Implements an N-dimensional version of the classic Perona-Malik anisotropic diffusion equation for scalar-valued images.

- The conductance term for this implementation is chosen as a function of the gradient magnitude of the image at each point, reducing the strength of diffusion at edge pixels.

$$C(\mathbf{x}) = e^{-\left(\frac{\|\nabla U(\mathbf{x})\|}{K}\right)^2}$$

2. Curvature Anisotropic Diffusion

- Performs anisotropic diffusion on an image using a modified curvature diffusion equation (MCDE).

- MCDE does not exhibit the edge enhancing properties of classic anisotropic diffusion, which can under certain conditions undergo a “negative” diffusion, which enhances the contrast of edges.

$$f_t = |\nabla f| \nabla \cdot c(|\nabla f|) \frac{\nabla f}{|\nabla f|}$$

MCDE equation

$$\nabla \cdot \frac{\nabla f}{|\nabla f|}$$

Conductance modified curvature term

3. Curvature Flow

- Performs edge-preserving smoothing in a similar fashion to the classical anisotropic diffusion.

- The filter uses a level set formulation where **the iso-intensity contours in an image are viewed as level sets,** where pixels of a particular intensity form one level set. The level set function is then evolved under the control of a diffusion equation where the **speed is proportional to the curvature** of the contour

$$I_t = \kappa|\nabla I|$$

3. Curvature Flow

- **High Curvature**: Areas with high curvature are regions where the iso-intensity contours are tightly curved, such as **around small noise artifacts**. These areas will diffuse (smooth) more quickly, which helps to remove small, jagged noise.

$$I_t = \kappa|\nabla I|$$

- **Low Curvature**: Areas with low curvature are **smoother and correspond to larger, more gradual changes in intensity**. These regions will evolve more slowly, which helps to preserve the sharp boundaries of significant features in the image, such as the edges of objects.

3. Curvature Flow

- Thus, continual application of this curvature flow scheme will eventually result in the removal of information as each contour shrinks to a point and disappears.

4. MinMaxCurvature Flow

- The MinMax curvature flow filter applies a variant of the curvature flow algorithm where diffusion is turned on or off depending of the scale of the noise that one wants to remove.

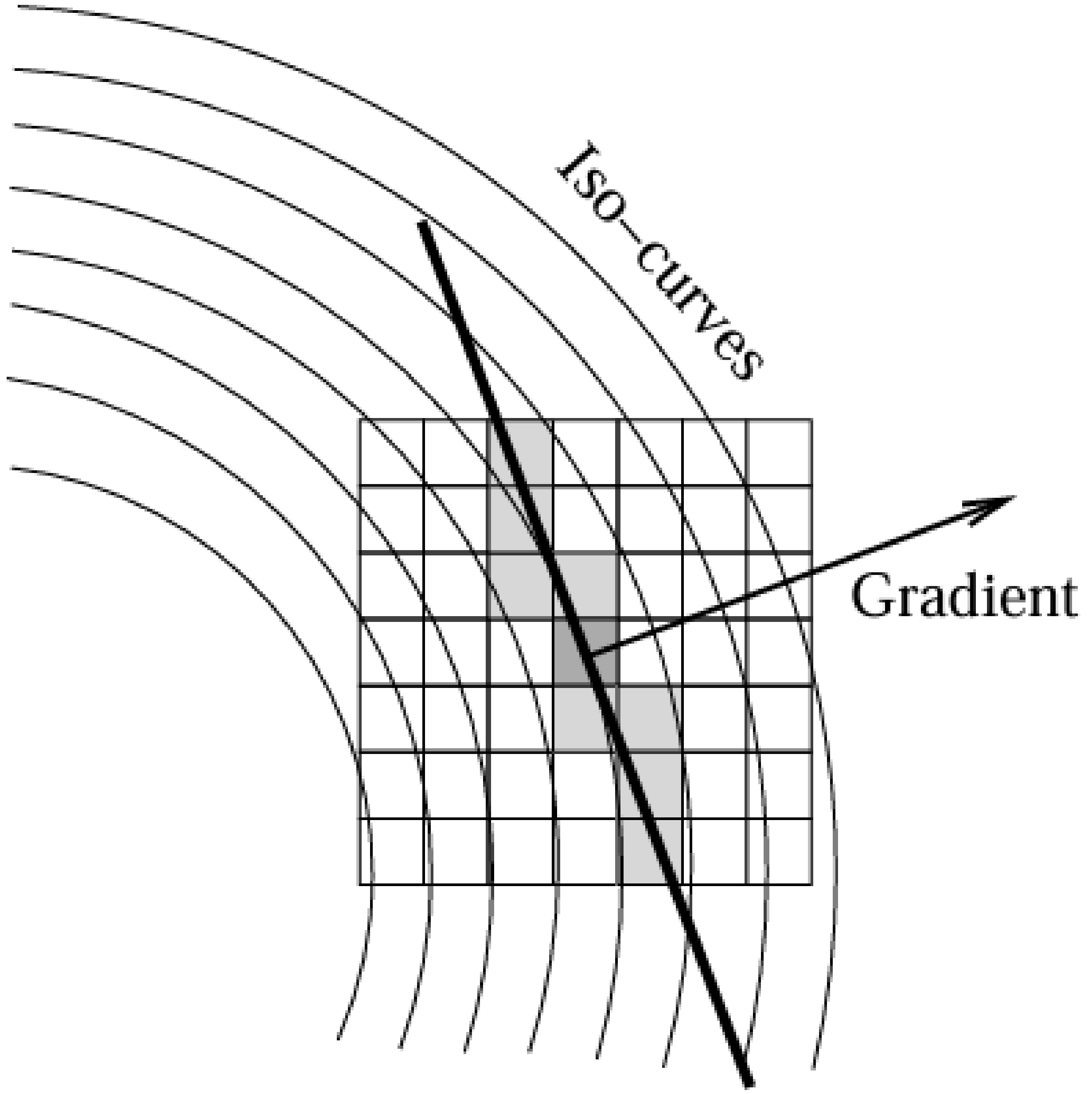

Figure 2.27: Elements involved in the computation of min-max curvature flow.

4. MinMaxCurvature Flow

- The Threshold is calculated as the average of pixel intensities along the direction perpendicular to the gradient at the extrema of the local neighborhood.

$$I_t = F|\nabla I|$$

$$F = \begin{cases} \max(\kappa, 0) & : \quad \text{Average} < Threshold \\ \min(\kappa, 0) & : \quad \text{Average} \geq Threshold \end{cases}$$

- The integer radius of the neighborhood is selected by the user.

4. MinMaxCurvature Flow

- F=max(κ,0) → This means that diffusion is turned on only in areas where the **curvature is positive**, i.e., where the local intensity curve is concave. This causes **small dark regions in a predominantly light area to shrink**, which is useful for removing dark noise spots.

- F=min(κ,0) → This applies diffusion where the **curvature is negative**, i.e., where the local intensity curve is convex. This causes **light regions in a predominantly dark area to shrink**, useful for removing bright noise spots in darker regions.

5. Bilateral Filter

- Performs smoothing by using both domain and range neighborhoods.

- Two Gaussian kernels (one in the image domain and one in the image range) are used to smooth the image. The result is an image that is smoothed in homogeneous regions yet has edges preserved. The result is similar to anisotropic diffusion but the implementation is non-iterative.

5. Bilateral Filter

$$h(\mathbf{x}) = k(\mathbf{x})^{-1} \int_{\omega} f(\mathbf{w}) c(\mathbf{x}, \mathbf{w}) s(f(\mathbf{x}), f(\mathbf{w})) d\mathbf{w}$$

f(x) : input image
h(x) : output image
c() : convolution kernels of spatial domain.
s() : convolution kernels of intensity domain.
ω : neighborhood of the pixel located at x in which the ND integral is calculated.
k(x) : normalization factor

$$k(\mathbf{x}) = \int_{\omega} c(\mathbf{x}, \mathbf{w}) s(f(\mathbf{x}), f(\mathbf{w})) d\mathbf{w}$$

$$c(\mathbf{x}, \mathbf{w}) = e^{\left(-\frac{\|\mathbf{x}-\mathbf{w}\|^2}{\sigma_c^2}\right)}$$

σ_c is provided by the user and defines how close pixel neighbors should be in order to be considered for the computation of the output value.

$$s(f(\mathbf{x}), f(\mathbf{w})) = e^{\left(-\frac{(f(\mathbf{x})-f(\mathbf{w}))^2}{\sigma_s^2}\right)}$$

σ_s is provided by the user and defines how close the neighbor’s intensity be in order to be considered for the computation of the output value.

Edge Preserving Smoothing in Vector Images

- Used for denoising data from devices that produce multiple values such as MRI or color photography.
- For processing registered data from different devices or for denoising higher order geometric or statistical features from scalar-valued images.
- These output images of reduced noise and texture but preserves, and can also enhance, edges, are useful for statistical classification, visualization, and geometric feature extraction.

1. Vector Gradient Anisotropic Diffusion

- Typically in vector-valued diffusion, vector components are diffused independently of one another using a conductance term that is linked across the components.

Vector Curvature Anisotropic Diffusion

- Performs anisotropic diffusion on a vector image using a modified curvature diffusion equation (MCDE).

- Typically in vector-valued diffusion, vector components are diffused independently of one another using a conductance term that is linked across the components.

Thank You!