Perona-Malik Diffusion Equation

The original Perona-Malik diffusion equation is given by:

$$\frac{\partial u}{\partial t} = \nabla \cdot (C(x, y, \|\nabla u\|)\nabla u)$$

Where:

- $u(x, y)$ represents the image intensity at spatial coordinates (x, y).
- t represents time, although in discretized form it represents the iteration step.
- $\nabla u = \left(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}\right)$ denotes the gradient of u.
- $\|\nabla u\| = \sqrt{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2}$ is the magnitude of the gradient.
- $C(x, y, \|\nabla u\|)$ is the diffusion coefficient which depends on the local image structure.

Discretized Form

In the discretized form used in the `perona_rhs` function, the equation is approximated as:

$$u^{t+1}(x, y) = u^t(x, y) + \Delta t \cdot \text{rhs}\left(u^t(x, y), \nu\right)$$

Where:

- $u^t(x, y)$ represents the image intensity at pixel (x, y) at time step t.
- $u^{t+1}(x, y)$ represents the updated image intensity at pixel (x, y) at time step $t + 1$.
- Δt is the time step or iteration step size.
- $\text{rhs}\left(u^t(x, y), \nu\right)$ denotes the right-hand side of the discretized equation, which is given by the expression calculated in `perona_rhs`.

Right-hand Side (rhs)

The right-hand side of the discretized equation in `perona_rhs` can be expressed as:

$$\text{rhs}\left(u(x, y), \nu\right) = \nabla \cdot (C(x, y, \|\nabla G_\sigma\|)\nabla u(x, y))$$

Where:

- $G_\sigma(x, y)$ is the image after applying a Gaussian filter with standard deviation σ.
- $\nabla\cdot$ denotes the divergence operator.
- $C(x, y, \|\nabla G_\sigma\|) = \exp\left(-\frac{\|\nabla G_\sigma\|^2}{\nu}\right)$ is the curvature coefficient which depends on the gradient magnitude $\|\nabla G_\sigma\|$ and the parameter ν.

Discrete Update Equation

The discrete update equation used in `perona_rhs` can be further expanded as:

$$\begin{aligned}
u^{t+1}(x, y) = u^t(x, y) + \Delta t \cdot \Bigg[&(2C(x, y, \|\nabla G_\sigma\|) + C(x + 1, y, \|\nabla G_\sigma\|)) \left(u^t(x + 1, y) - u^t(x, y)\right) \\
&- (2C(x - 1, y, \|\nabla G_\sigma\|) + C(x, y, \|\nabla G_\sigma\|)) \left(u^t(x, y) - u^t(x - 1, y)\right) \\
&+ (2C(x, y, \|\nabla G_\sigma\|) + C(x, y + 1, \|\nabla G_\sigma\|)) \left(u^t(x, y + 1) - u^t(x, y)\right) \\
&- (2C(x, y - 1, \|\nabla G_\sigma\|) + C(x, y, \|\nabla G_\sigma\|)) \left(u^t(x, y) - u^t(x, y - 1)\right) \Bigg]
\end{aligned}$$

This equation describes how the image intensity u at each pixel is updated based on the local image structure (through the gradient magnitude and curvature coefficient C) and the differences between neighboring pixels.

Discretization of the Proposed Model

In order to present a numerical approximation of the fractional derivative of Caputo, we set $\Omega(x_i, y_j)$ as the spatial partition of the image u for all $i = 1, \ldots, N$ and $j = 1, \ldots, M$. We denote $u_{i,j}$ as the value of the image u at the pixel (inner point) (i, j). Let τ be the time step, i.e., $t_0 = 0$, $t_{\max} = T$, and $t_k = k\tau$ for $k = 1, 2, \ldots, T_{\max}$.

The Caputo fractional derivative of u at the inner point (i, j) is approached by [?]:

$$\mathcal{D}_c^\alpha u_{k,i,j} \approx \sigma^{\alpha,\tau} \sum_{l=1}^{k} \omega_l^\alpha (u_{k-l+1,i,j} - u_{k-l,i,j}) = \sigma^{\alpha,\tau} \left[u_{k,i,j} - \sum_{l=1}^{k-1} (\omega_l^\alpha - \omega_{l+1}^\alpha) u_{k-l,i,j} - \omega_k^\alpha u_{0,i,j} \right],$$

where

$$\sigma^{\alpha,\tau} = \frac{\tau^{-\alpha}}{\Gamma(2-\alpha)}, \quad \omega_l^\alpha = l^{1-\alpha} - (l-1)^{1-\alpha}, \quad \text{and} \quad 1 = \omega_1^\alpha > \omega_2^\alpha > \cdots > \omega_k^\alpha.$$

Now, we approximate the term $\operatorname{div}(\zeta_{k,i,j} \nabla u_{k,i,j})$, where $\zeta_{k,i,j} = \zeta_{k,i,j}(|\nabla(u_{k,i,j})|\tau)$. We first define the following classical discrete approximation:

$$\nabla_x^+ u_{k,i,j} = u_{k,i+1,j} - u_{k,i,j}, \quad \nabla_x^- u_{k,i,j} = u_{k,i,j} - u_{k,i-1,j},$$

$$\nabla_y^+ u_{k,i,j} = u_{k,i,j+1} - u_{k,i,j}, \quad \nabla_y^- u_{k,i,j} = u_{k,i,j} - u_{k,i,j-1}.$$

The discrete approximation of the divergence operator is computed as:

$$\mathrm{div}(\zeta_{k,i,j} \nabla u_{k,i,j}) = \nabla_x^- (\zeta_{k,i,j} \nabla_x^+ u_{k,i,j}) + \nabla_y^- (\zeta_{k,i,j} \nabla_y^+ u_{k,i,j}).$$

The final discretization of our proposed model, for $1 \leq k \leq T_{\max}$, is expressed as:

$$u_{k,i,j} = \sum_{l=1}^{k-1} (\omega_l^\alpha - \omega_{l+1}^\alpha) u_{k-l,i,j} - \omega_k^\alpha u_{0,i,j} + \tau^\alpha \frac{\Gamma(2-\alpha)}{\Gamma(1-\alpha)} \left[\nabla_x^- (\zeta_{k-1,i,j} \nabla_x^+ u_{k-1,i,j}) + \nabla_y^- (\zeta_{k-1,i,j} \nabla_y^+ u_{k-1,i,j}) \right].$$

Constructed Difference Scheme and Its Stability

Let us introduce grids with uniform steps that are given as

$$W_h = \{x_n : x_n = nh,\ n = 0, 1, \ldots, M\}, \quad h = \frac{X}{M},$$

$$W_\tau = \{t_k : t_k = k\tau,\ k = 0, 1, \ldots, N\}, \quad \tau = \frac{T}{N}.$$

The first-order difference scheme is given by

$$D_t^\sigma (u(t_k, x_n)) = \frac{1}{\Gamma(\sigma)} \sum_{j=0}^{k} \frac{u_{k+1,n} - u_{k,n}}{\tau} d_{j,k},$$

where

$$d_{j,k} = (t_j - t_{k+1})^{1-\alpha} - (t_j - t_k)^{1-\alpha}.$$

The difference scheme is provided as

$$D_t^\sigma (u(t_k, x_n)) = \tau^{-\alpha} \frac{1}{\Gamma(2-\sigma)} \sum_{j=0}^{k} w_j^\sigma (u_{k-j+1,n} - u_{k-j,n}),$$

where

$$w_j^\sigma = (j+1)^{1-\sigma} - j^{1-\sigma}.$$

Using Taylor expansion, the Dufort–Frankel difference formula for $u_{xx}(t_k, x_n)$ is given by

$$u_{xx}(t_k, x_n) \approx \frac{u_{k,n+1} - (u_{k-1,n} + u_{k+1,n}) + u_{k,n-1}}{h^2}.$$

From, the second-order difference approximation for $u_{tt}(t_k, x_n)$ is expressed as

$$u_{tt}(t_k, x_n) \approx \frac{1}{\tau^2} (u_{k+1,n} - 2u_{k,n} + u_{k-1,n}).$$

$$U_{tt} + \gamma U_t^\alpha = \text{Perona}$$

$$\frac{U_n^{k+1} - 2U_n^k + U_n^{k-1}}{\tau^2} + \gamma \left[\sum_{i=1}^{k} \psi_{i,j}^k \left(\omega_l^k - \omega_{l+1}^k \right) \psi_l^k \right] + \gamma \tau^2 (\text{Perona})$$

$$\nu = \text{constant}, \quad \gamma = \text{constant}, \quad \alpha = \text{constant parameter}$$

heloo

$$U_n^{k+1} = 2U_n^k - U_n^{k-1} - \gamma \tau^2 \left[U_n^k - \sum_{l=1}^{k-1} (\omega_l - \omega_{l+1}) U_l \right] - \omega^\alpha U_0^\alpha + \gamma \tau^2 (\text{Perona})$$

$$U^{k+1} = U^k \left[2 - \gamma \tau^2 - 2 \right] + \gamma \tau^2 \left[\sum_{l=1}^{k-1} (\omega_l - \omega_{l+1}) U^{k-1} \right] - \omega^\alpha U_0^\alpha + \gamma \tau^2 (\text{Perona})$$