SHAPING REGULARIZATION

Sergey Fomel
The University of Texas at Austin

SIAM Conference on Mathematical & Computational Issues in the Geosciences

SHAPING REGULARIZATION

S. Fomel, 2007, Shaping regularization in geophysical-estimation problems: Geophysics.

- General framework for inversion
- Connection with sparsity
- Triangle shaping by fast explicit diffusion

MOST POPULAR PROGRAMS

- sfmath
- sfwindow
- sfput
- sfdd
- sfspike
- sfcat
- sfadd
- sftransp
- sfscale
- sfsmooth
- sfspray
- sfricker1
- sfstack
- sfpad
- sfnoise
- sfsegyread
- sfcut
- sfcmplx
- sfdip
- sfrtoc
- sfmutter
- sfenvelope
- sfpick
- sfclip2
- sfreverse
- sfpow
- sfintbin

GENERAL FRAMEWORK

$$\mathbf{d} = \mathbf{F}\,\mathbf{m}$$

$$\tilde{\mathbf{m}} = \mathbf{B}\,\mathbf{d}$$

$$\mathbf{m}_{n+1} = \mathbf{m}_n + \tilde{\mathbf{m}} - \mathbf{B}\,\mathbf{F}\,\mathbf{m}_n = \tilde{\mathbf{m}} + (\mathbf{I} - \mathbf{B}\,\mathbf{F})\,\mathbf{m}_n$$

$$\mathbf{m}_{n+1} = \mathbf{S}\left[\tilde{\mathbf{m}} + (\mathbf{I} - \mathbf{B}\,\mathbf{F})\,\mathbf{m}_n\right]$$

SHAPING ITERATION

- Different from optimization
- Two goals: fitting data and having shape

CONNECTION WITH OPTIMIZATION

$$\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left(\|\mathbf{L}\mathbf{x} - \mathbf{d}\|^2 + \epsilon^2 \|\nabla \mathbf{x}\|^2 \right)$$

SHAPING FOR SMOOTHNESS

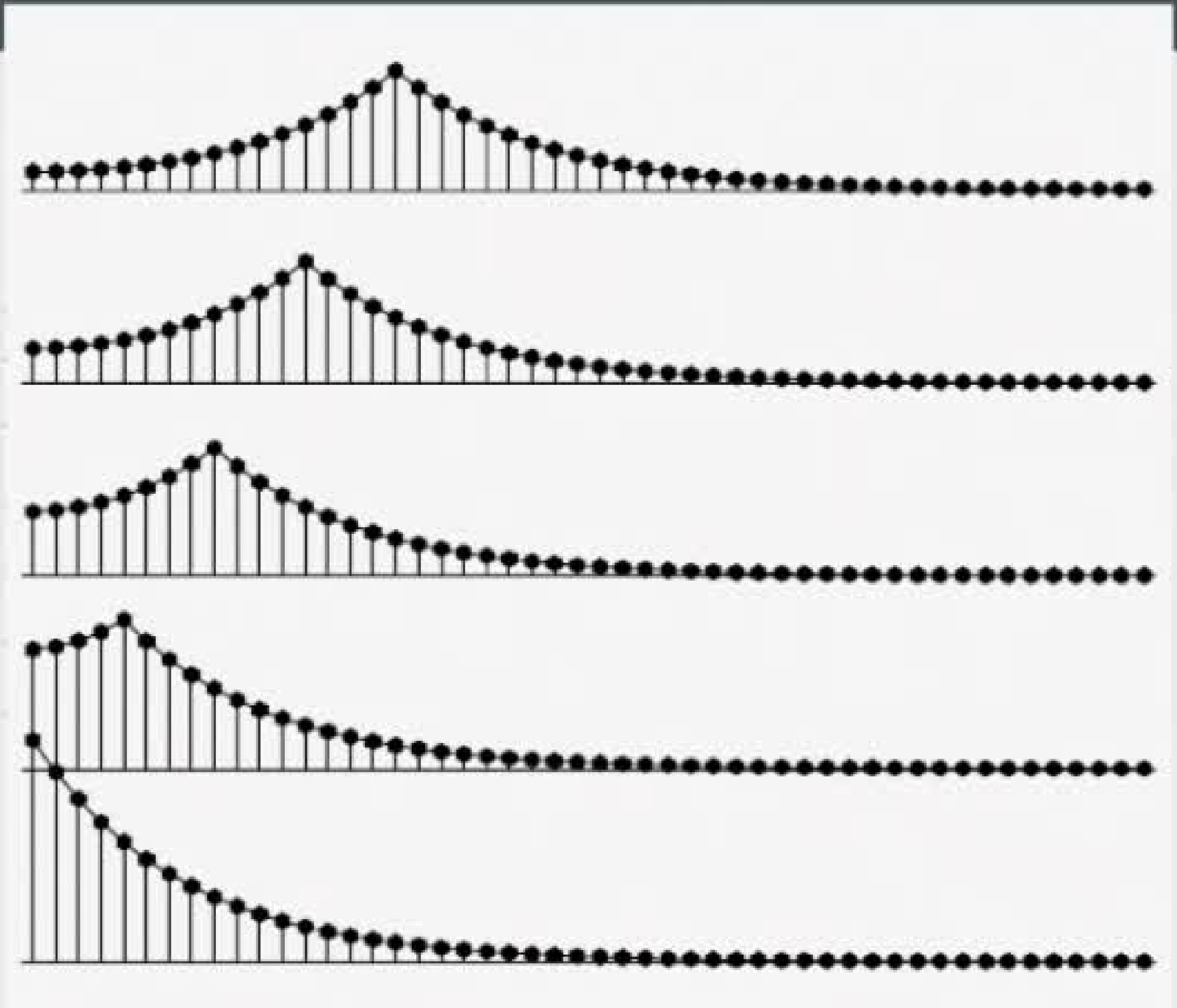
SHAPING FOR SMOOTHNESS

$$P[\mathbf{x}] = \epsilon^2 \left\| \nabla \, \mathbf{x} \right\|^2$$

$$S[\mathbf{x}] = (\mathbf{I} - \epsilon^2 \mathbf{D})^{-1}$$

CLAERBOUT (1992)

- "I habitually smoothed with damped exponentials, but I switched to triangles after I encountered several examples where the exponential tails decreased too slowly."



SHAPING WITH TRIANGLE SMOOTHING VS. PENALIZING GRADIENTS

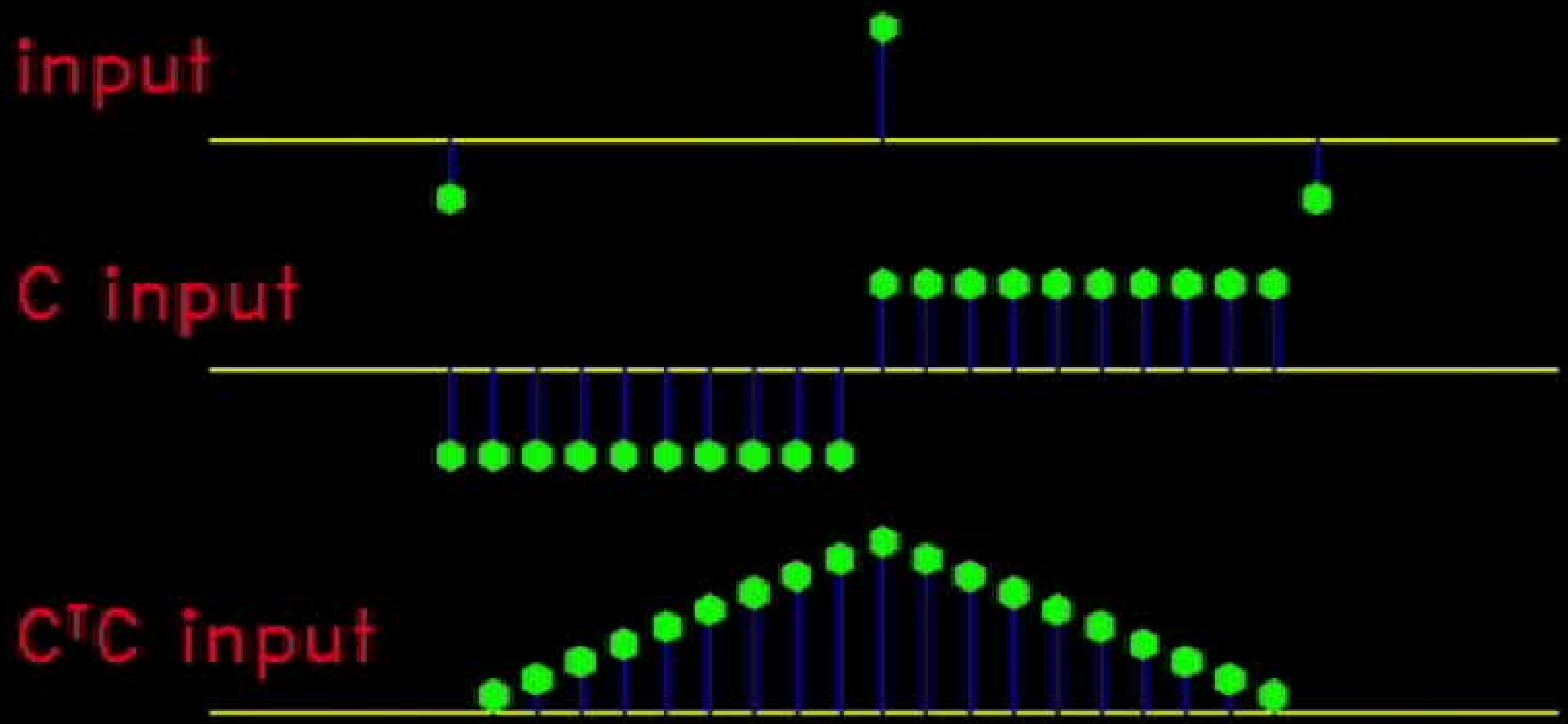
Significantly faster iterative convergence

Intuitive selection of parameters

Stronger regularization leads to faster convergence

FROM TRIANGLE SMOOTHING TO FAST EXPLICIT DIFFUSION

TRIANGLE SMOOTHING (CLAERBOUT, 1992)



$$B_N(Z) = \frac{1}{N}\left(1 + Z + Z^2 + \cdots + Z^{N-1}\right) = \frac{1 - Z^N}{N\,(1 - Z)}$$

$$T_N(Z) = B_N(Z)\,B_N(1/Z) = \frac{\left(1 - Z^N\right)\left(1 - Z^{-N}\right)}{N^2\,(1 - Z)\,(1 - 1/Z)}$$

BOX TO TRIANGLE Z-TRANSFORM

- Recursive implementation
- Minimal cost independent of the filter size

$$B_N(Z) = \frac{1 - Z^N}{N\,(1 - Z)}$$

$$1 - Z^N = \prod_{n=0}^{N-1} \left(1 - \frac{Z}{Z_n}\right)$$

$$Z_n = \exp\left(i\,\frac{2\pi\,n}{N}\right)$$

FACTORING BOX FILTER

- Non-recursive implementation
- Cascade of elementary filters

$$= \frac{1}{2N+1} \prod_{n=1}^{N} \left(1 - 2\,\gamma_n\, Z + Z^2\right)$$

$$\gamma_n = \cos\left(\frac{2\pi\, n}{2N+1}\right)$$

FACTORED BOX

- Polynomial roots come in conjugate pairs
- Cascade of elementary three-point filters

FACTORED TRIANGLE

$$T_N(Z) = \frac{1}{N^2} \prod_{n=1}^{N-1} (Z + 1/Z - 2\,\gamma_n)$$

$$= \frac{1}{2N+1} \prod_{n=1}^{N} \left(1 - 2\,\gamma_n\, Z + Z^2\right)$$

$$\gamma_n = \cos\left(\frac{2\pi\, n}{2N+1}\right)$$

FACTORED BOX

- Polynomial roots come in conjugate pairs
- Cascade of elementary three-point filters

FACTORED TRIANGLE

$$T_N(Z) = \frac{1}{N^2} \prod_{n=1}^{N-1} (Z + 1/Z - 2\,\gamma_n)$$

CONNECTION WITH DIFFUSION

$$F_B(\omega) = \frac{1}{2b} \int_{-b}^{b} e^{-i\omega t}\, dt = \frac{1}{2i\omega b} \left(e^{i\omega b} - e^{-i\omega b}\right) = \frac{\sin(\omega b)}{\omega b} = \mathrm{sinc}(\omega b)$$

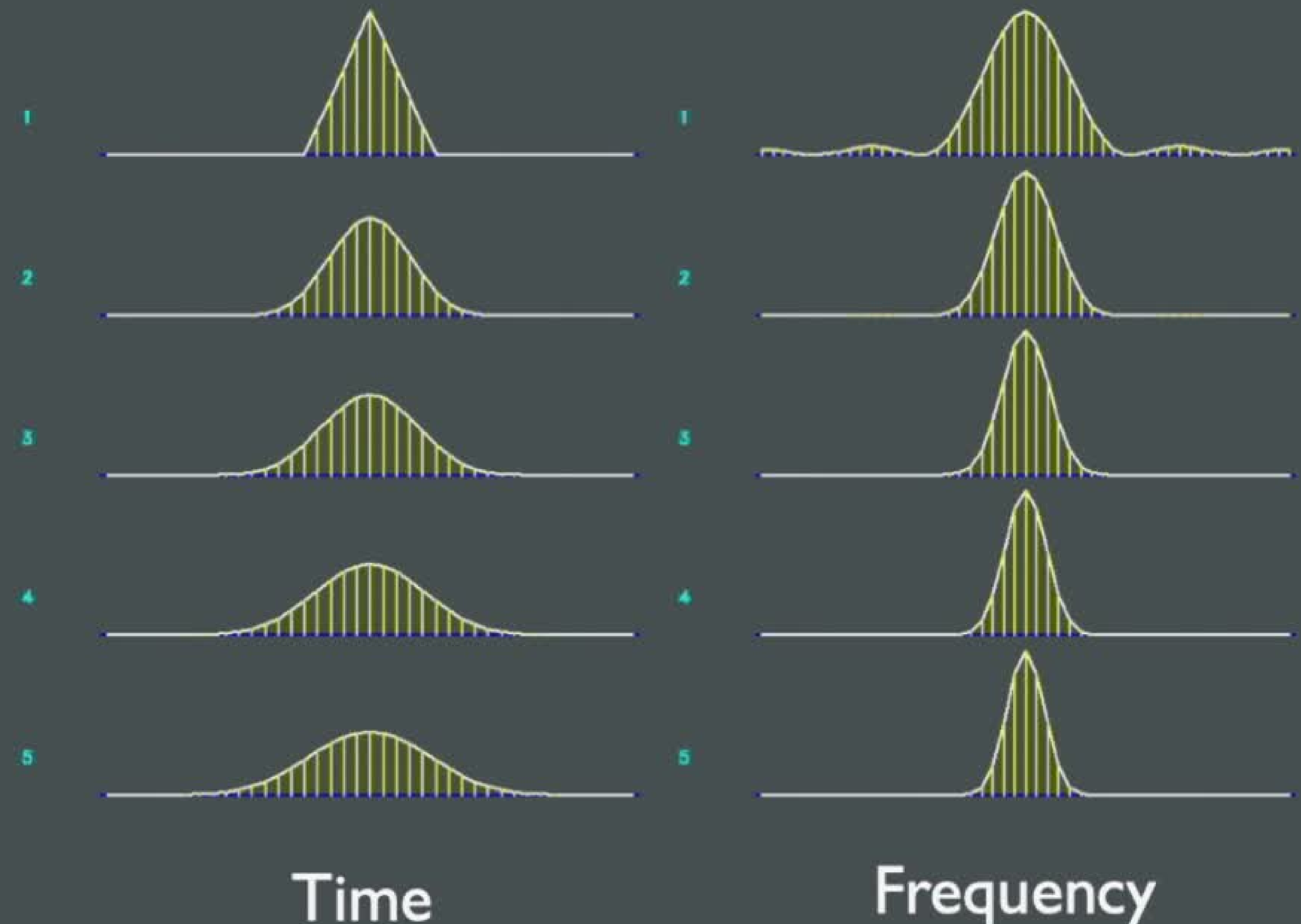
$$F_T(\omega) = |F_B(\omega)|^2 = \mathrm{sinc}^2(\omega b)$$

$$\mathrm{sinc}^n(\omega b) \approx e^{-\omega^2 b^2 n/6}$$

WHY TRIANGLE?

- Converges to a Gaussian (ideal smoother)
- As follows from the central limit theorem

Convergence to a Gaussian



$$U(k,t) = U(k,0)\,\exp(-a^2\,t\,k^2)$$

$$u(x,t) = \frac{1}{\sqrt{4\pi\,a^2\,t}} \int u_0(x_0)\,\exp\left(-\frac{|x-x_0|^2}{4\,a^2\,t}\right)\,dx_0$$

DIFFUSION PROCESS

- Physical process
- Leads to Gaussian smoothing

$$\frac{\partial u}{\partial t} = a^2 \frac{\partial^2 u}{\partial x^2}$$

$$\frac{\partial \mathbf{u}}{\partial t} = a^2 \, \mathbf{D} \, \mathbf{u}$$

$$\frac{\mathbf{u}_{n+1} - \mathbf{u}_n}{\Delta t} = a^2 \, \mathbf{D} \, \mathbf{u}$$

NUMERICAL DIFFUSION

- Discretize in space
- Discretize in time

$$\mathbf{u}_{n+1} = \left(\mathbf{I} + a^2\, \Delta t\, \mathbf{D}\right)\, \mathbf{u}_n$$

$$\mathbf{u}_{n+1} = \left(\mathbf{I} - a^2\, \Delta t\, \mathbf{D}\right)^{-1}\, \mathbf{u}_n$$

EXPLICIT VS. IMPLICIT

- Explicit: requires tiny steps for stability
- Implicit: more stable, but requires inversion

SHAPING FOR SMOOTHNESS

$$P[\mathbf{x}] = \epsilon^2 \left\| \nabla \, \mathbf{x} \right\|^2$$

$$\mathbf{S}[\mathbf{x}] = \left(\mathbf{I} - \epsilon^2 \, \mathbf{D}\right)^{-1}$$

$$\mathbf{u}_{n+1} = \left(\mathbf{I} + a^2\,\Delta t\,\mathbf{D}\right)\,\mathbf{u}_n$$

$$\mathbf{u}_{n+1} = \left(\mathbf{I} - a^2\,\Delta t\,\mathbf{D}\right)^{-1}\,\mathbf{u}_n$$

EXPLICIT VS. IMPLICIT

- Explicit: requires tiny steps for stability
- Implicit: more stable, but requires inversion

SHAPING FOR SMOOTHNESS

$$P[\mathbf{x}] = \epsilon^2 \left\| \nabla \mathbf{x} \right\|^2$$

$$\mathbf{S}[\mathbf{x}] = \left(\mathbf{I} - \epsilon^2 \mathbf{D}\right)^{-1}$$

$$G_\alpha(Z) = \prod_{n=1}^{N-1} \left[1 + \alpha\,(Z + 1/Z - 2)\right]$$

$$\alpha = \frac{a^2\,\Delta t}{(\Delta x)^2}$$

EXPLICIT DIFFUSION FILTER

- Three-point Laplacian discretization
- Cascade of elementary three-point filters

FACTORED TRIANGLE

$$T_N(Z) = \frac{1}{N^2} \prod_{n=1}^{N-1} (Z + 1/Z - 2\,\gamma_n)$$

$$= \prod_{n=1}^{N-1} \left(1 + \frac{Z + 1/Z - 2}{\alpha_n}\right)$$

GENERALIZED DIFFUSION

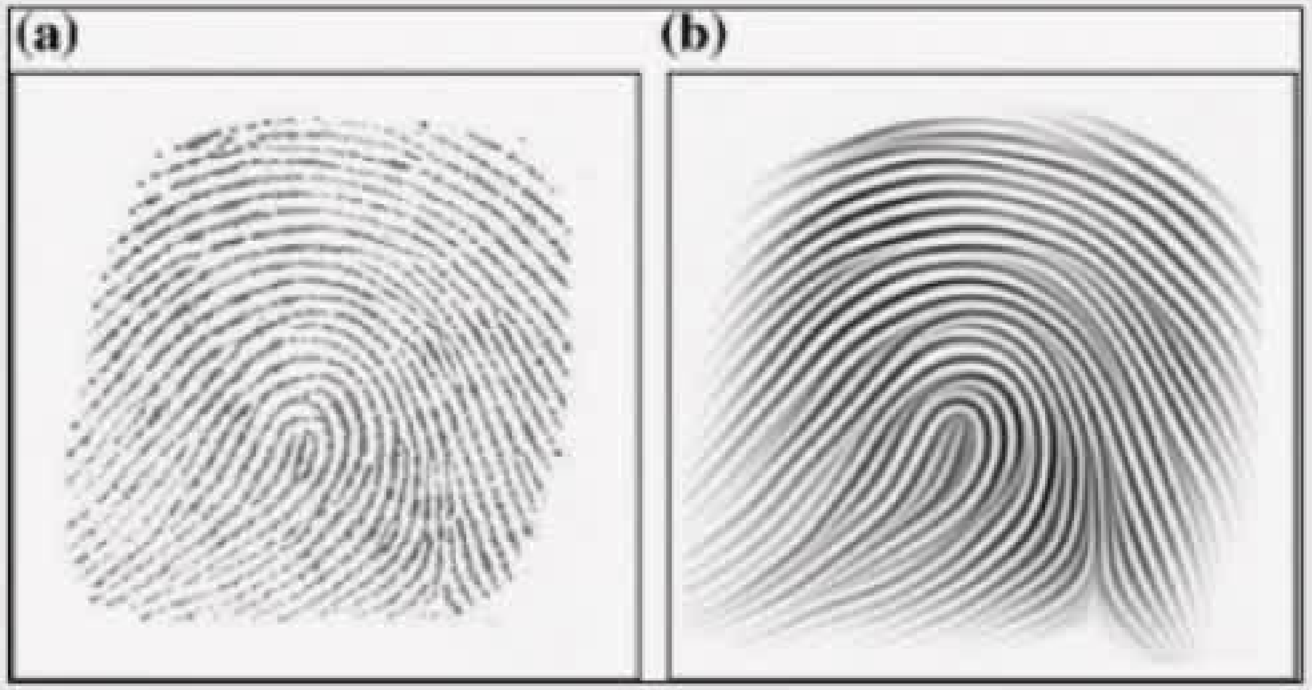
$$\frac{\partial \mathbf{u}}{\partial t} = a^2 \, \mathbf{D} \, \mathbf{u}$$

↓

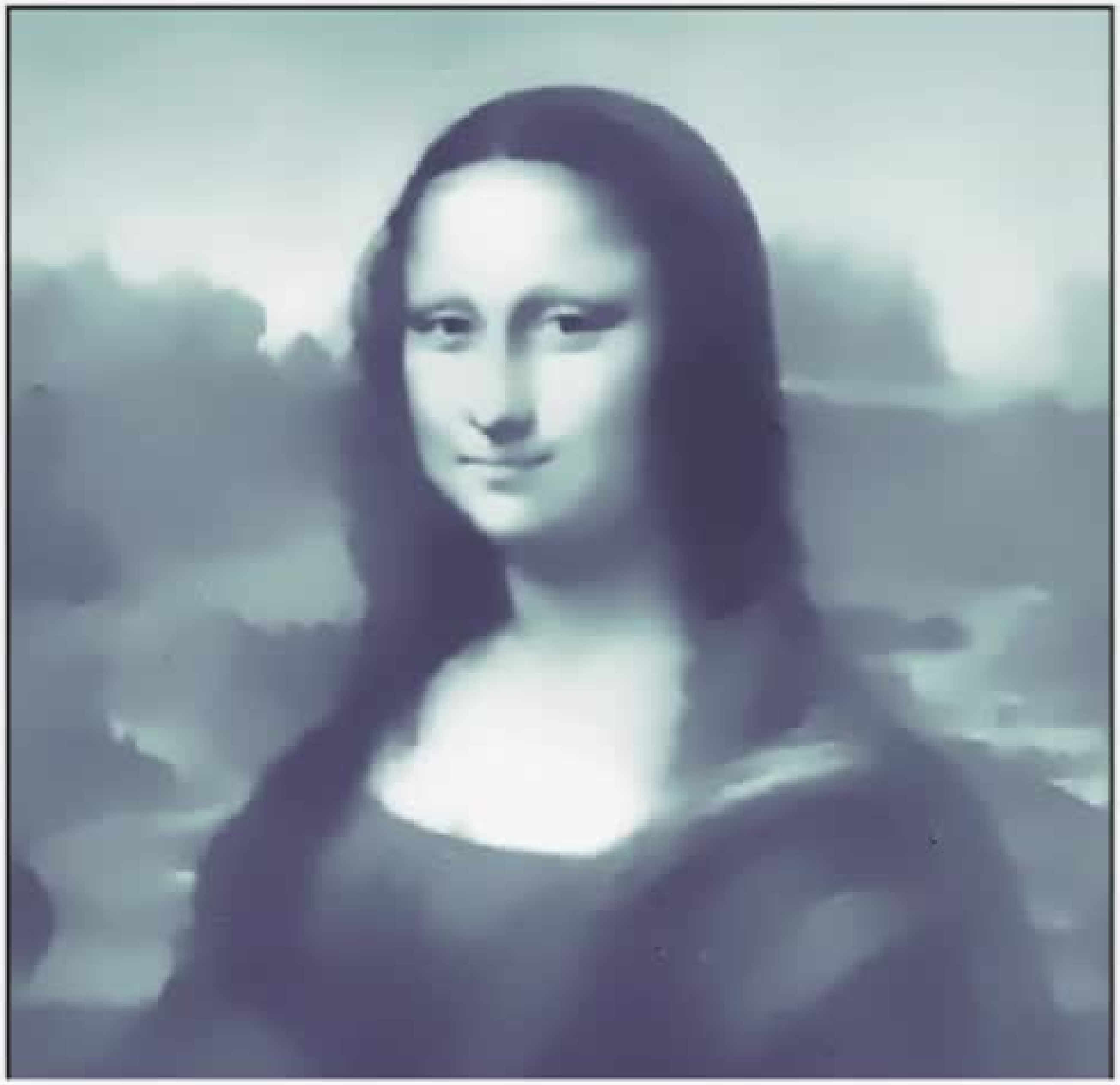
$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{G}[\mathbf{u}]$$

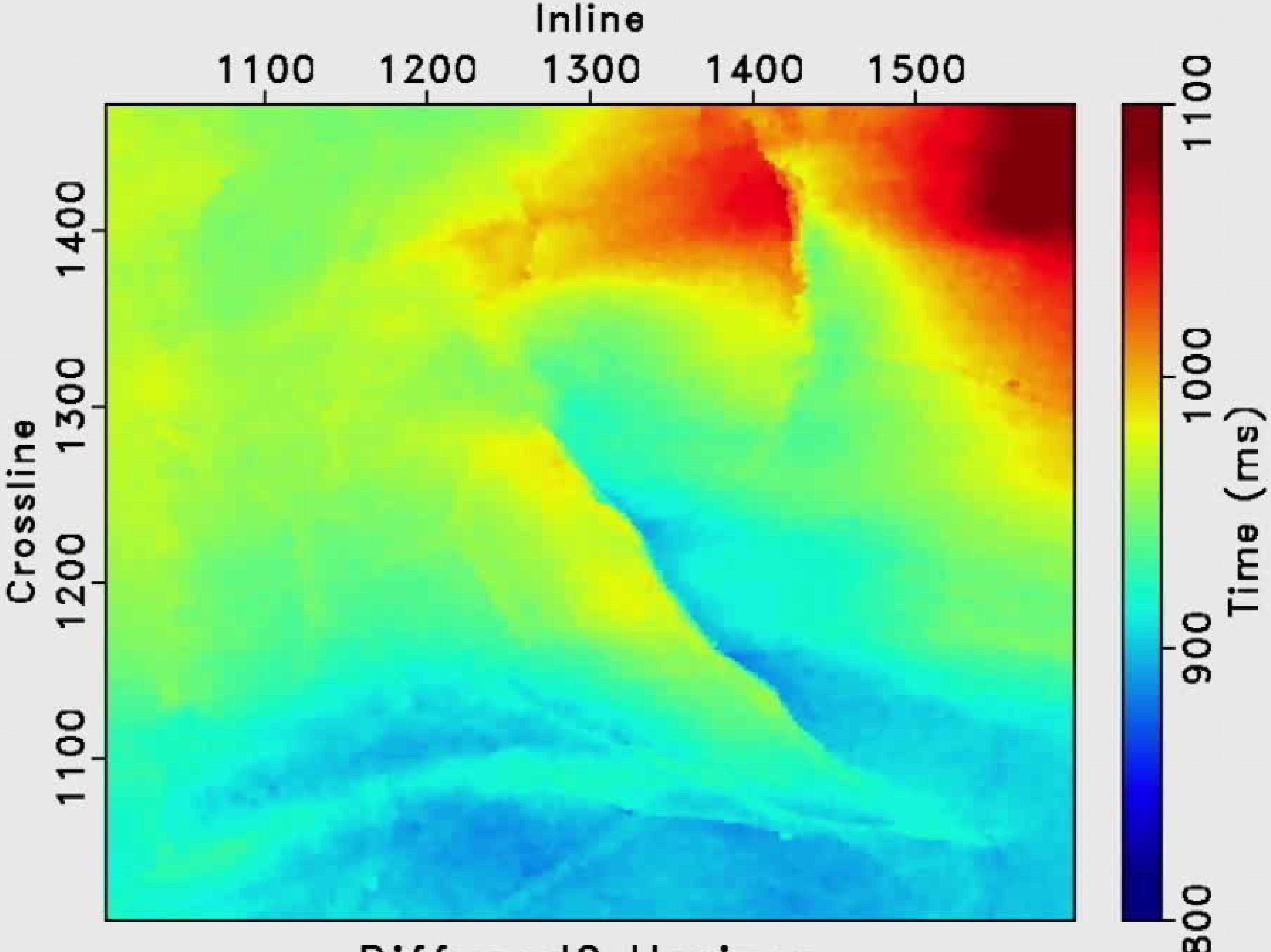
FED (FAST EXPLICIT DIFFUSION)

$$\mathbf{u}_{n+1} = (\mathbf{I} + \Delta t_n \, \mathbf{G}) \; \mathbf{u}_n$$

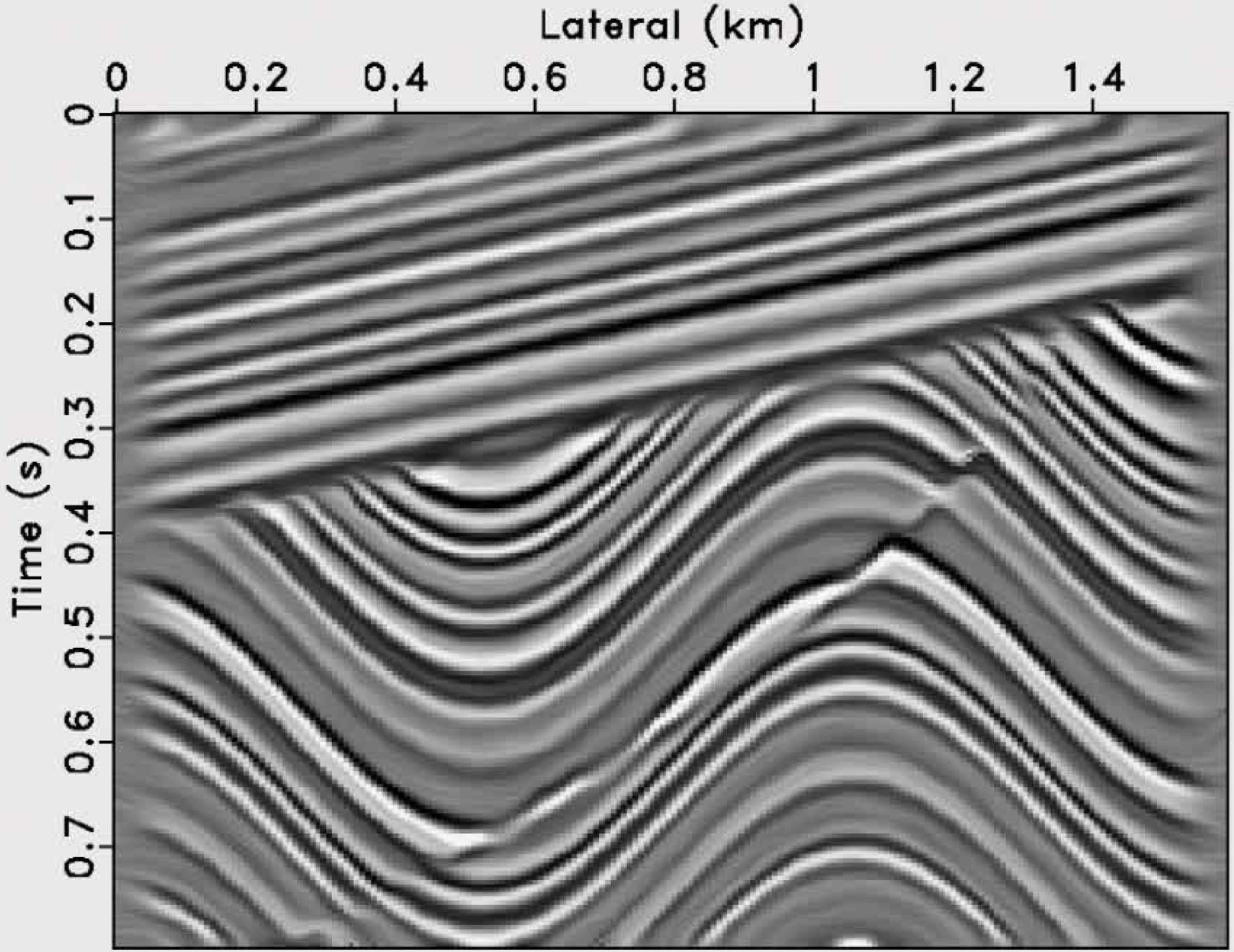


J. Weickert, S. Grewenig, C. Schroers, and A. Bruhn, 2016. Cyclic schemes for PDE-based image analysis. International Journal of Computer Vision, *118*, 275-299.

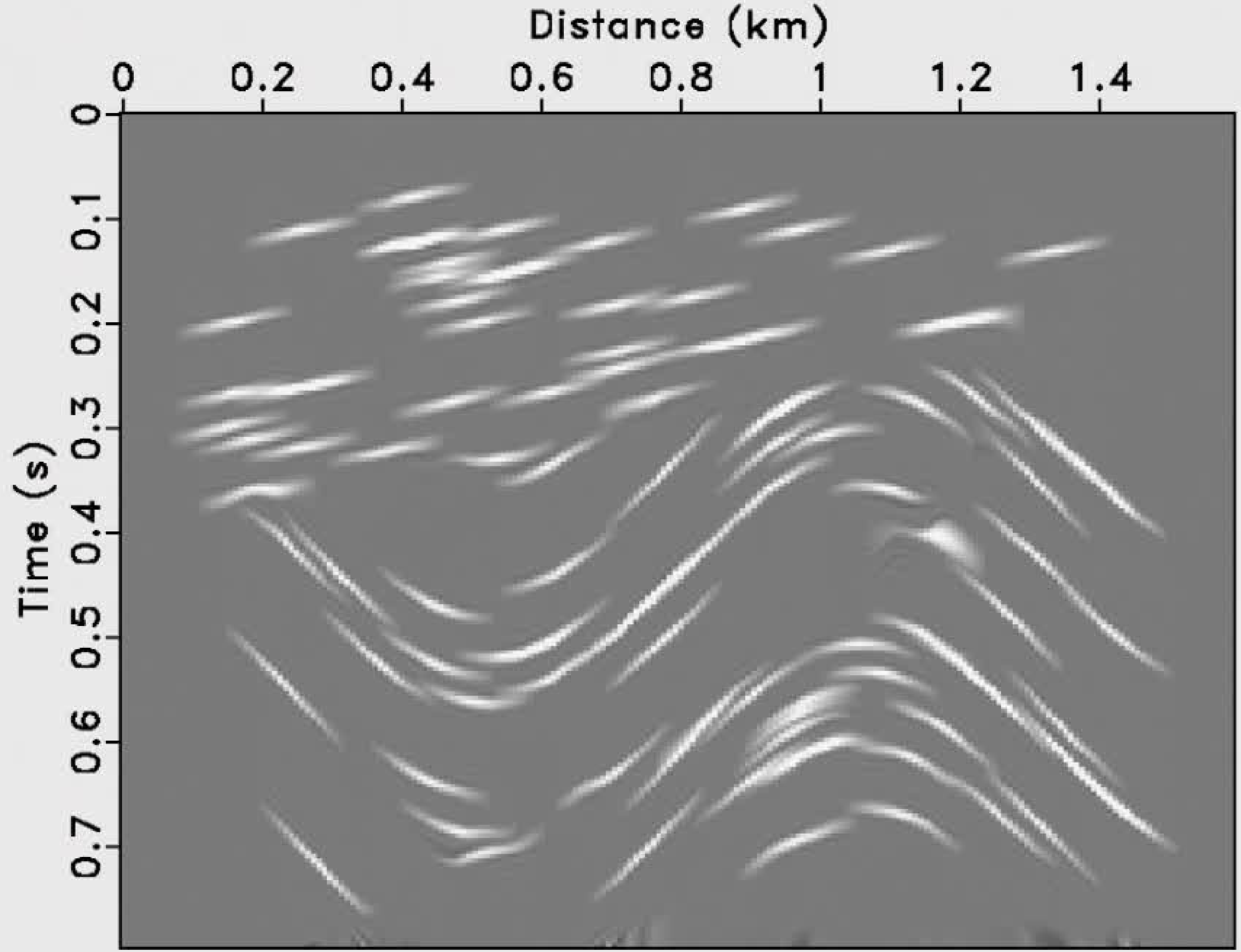




Diffused2 Horizon



Smoothed



Structure-Oriented Smoothing