Gabor Depth Imaging and Adaptive Windowing Algorithms

Yongwang Ma and Gary F. Margrave

December 2, 2005

Outline

Motivations

- Accurate imaging
- Fast imaging
- Trading off between accuracy and efficiency
- Developing depth imaging algorithms for complex 2D and 3D velocity models

The Continuous Gabor Transforms

(after Margrave et al, 2001)

$$V_g s(x'_T, k_T) = \int_{\mathbb{R}} s(x_T) g(x_T - x'_T) e^{-ix_T k_T} \mathrm{d}x_T \tag{1}$$

$$s(x_T) = \int_{\mathbb{R}^2} V_g s(x'_T, k_T) \gamma(x_T - x'_T) e^{ix_T k_T} \mathrm{d}k_T \mathrm{d}x'_T \tag{2}$$

$$\int_{\mathbb{R}} g(x_T)\gamma(x_T)\mathrm{d}x_T = 1 \tag{3}$$

$s(x_T)$: signal

$V_g s(x'_T, k_T)$: Gabor spectrum of $s(x_T)$

x_T: transverse coordinate

k_T: transverse wavenumber

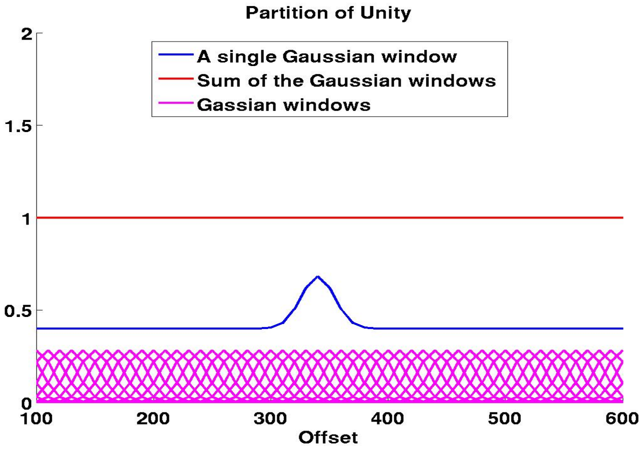
$g(x_T - x'_T)$: analysis window

$\gamma(x_T - x'_T)$: synthesis window

x'_T: window centre

$\mathbb{R}$: real domain

Partition of Unity



The GPSPI Extrapolation

(Margrave and Ferguson,1999; Margrave et al,2004)

$$\psi_P(x_T, z+\Delta z, \omega) = \int_{\mathbb{R}} \hat{\psi}(k_T, z, \omega)\hat{W}(k_T, x_T, \Delta z)e^{-ik_T x_T}\mathrm{d}k_T \tag{4}$$

$$\hat{W}(k_T, x_T, \Delta z) = e^{ik_z \Delta z} \tag{5}$$

$$k_z(x_T) = \begin{cases} \sqrt{\frac{\omega^2}{v^2(x_T)} - k_T^2}, & \frac{\omega^2}{v^2(x_T)} > k_T^2 \\ i\sqrt{k_T^2 - \frac{\omega^2}{v^2(x_T)}}, & \frac{\omega^2}{v^2(x_T)} < k_T^2 \end{cases} \tag{6}$$

Approximation of the GPSPI Extrapolation

$$\hat{W}(k_T, x_T, \Delta z) \approx \sum_{j \in \mathbb{Z}} \Omega_j(x_T) S_j(x_T) \hat{W}(k_T, \Delta z) \tag{7}$$

$$\begin{aligned} \psi_P(x_T, z + \Delta z) \quad \approx \quad & \sum_{j \in \mathbb{Z}} \Omega_j(x_T) S_j(x_T) \int_{\mathbb{R}} \hat{\psi}(k_T, z, \omega) \\ & \cdot \hat{W}(k_T, \Delta z) e^{-i k_T x_T} \mathrm{d}k_T \end{aligned} \tag{8}$$

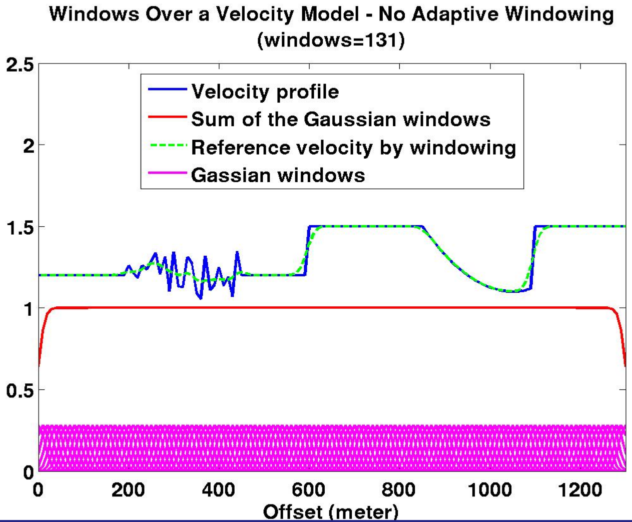
Some Definitions

$$S_j(x_T) = e^{i\omega\Delta z\left(\frac{1}{v(x_T)} - \frac{1}{v_j}\right)} \tag{9}$$

$$\hat{W}(k_T, \Delta z) = e^{ik_z\Delta z} \tag{10}$$

$$v_j = \frac{\int_{\mathbb{R}} \Omega_j(x_T) v(x_T) \mathrm{d}x_T}{\int_{\mathbb{R}} \Omega_j(x_T) \mathrm{d}x_T} \tag{11}$$

Windows - No Adaptive Windowing Algorithm



Phase Error Adaptive Windowing (PEAW) Algorithm

$$\epsilon_{jr} = \frac{\|A - B\|_1}{\|A\|_1}, \tag{12}$$

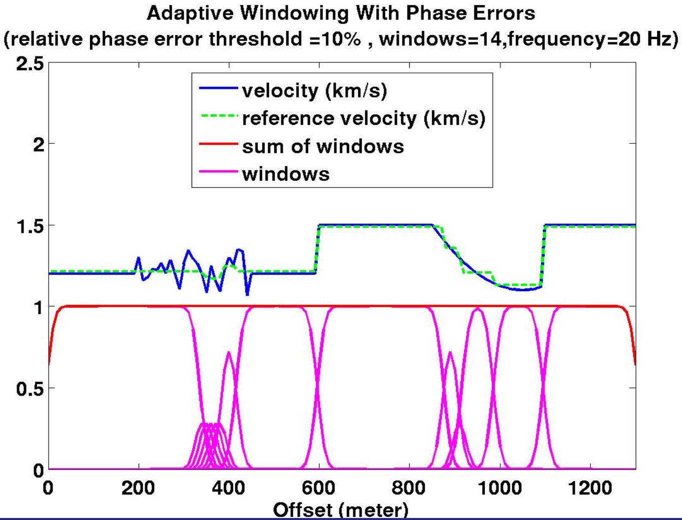
where

$$A = \arg\left(\Omega_j(x_T)\hat{W}(k_T, x_T, \Delta z)\right) \tag{13}$$

$$B = \arg\left(\Omega_j(x_T)\sum_{k\in\mathbb{Z}}\Omega_k(x_T)S_k(x_T)\hat{W}_k(k_T, \Delta z)\right) \tag{14}$$

$$j, k \in \mathbb{Z}$$

Windows - the PEAW Algorithm

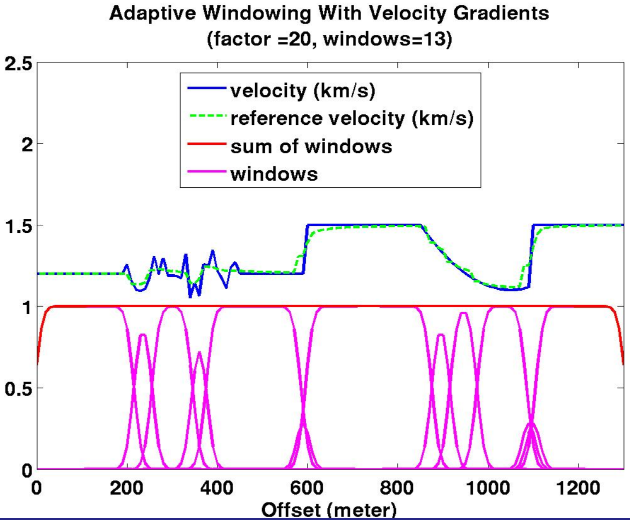


Velocity Gradient Adaptive Windowing (VGAW) Algorithm (Grossman et al, 2001)

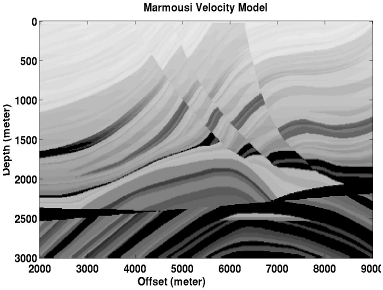
$$M_n^N(x) = \sum_{j=n_1}^{n_N} g(x - x_j), \quad v_{M_n^N} = \frac{\sum_{j=n_1}^{n_N} v(x_j) M_n^N(x_j)}{\sum_{j=n_1}^{n_N} M_n^N(x_j)} \tag{15}$$

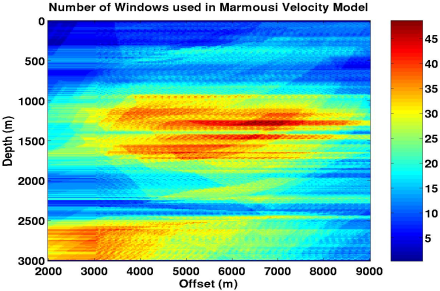
if $\|v(x_N + 1) - v_{M_n^N}\| < \lambda$, then
$M_n^N \rightarrow M_n^{N+1} = m_n^N + g_{n_N+1}$,
otherwise, $M_n^N \rightarrow M_n^N$ and $M_{n+1}^1 = g_{n_N+1}$

Windows - the VGAW Algorithm

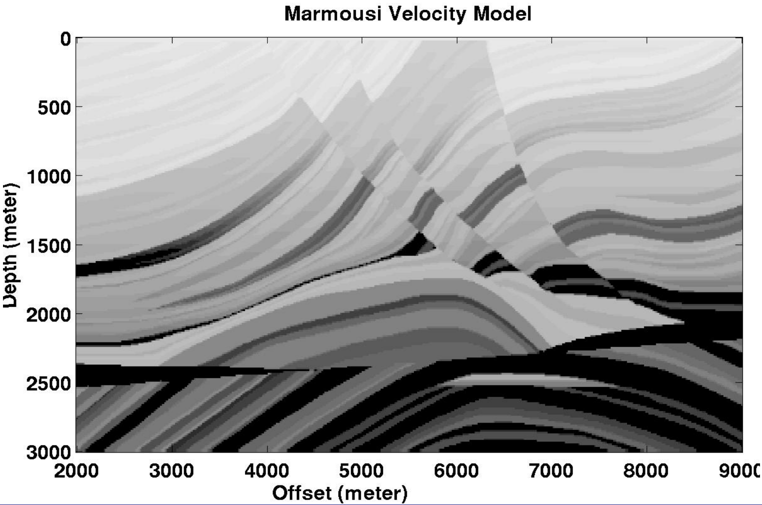


Windows Used in Marmousi Velocity Model (factor=20)

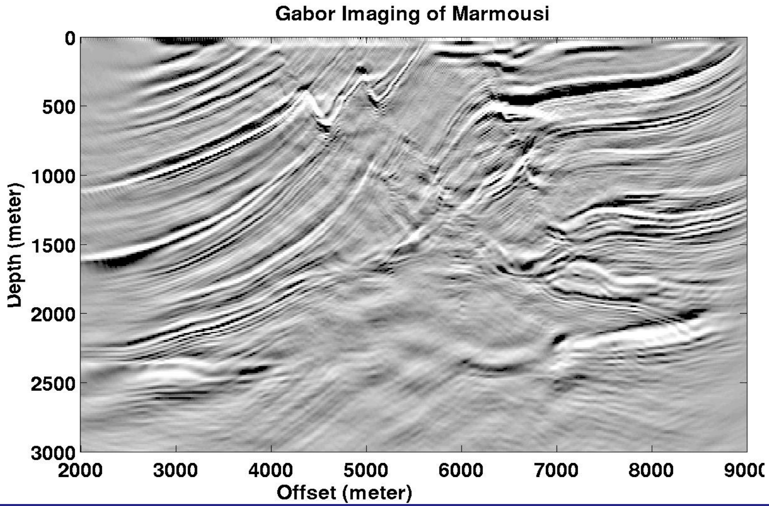




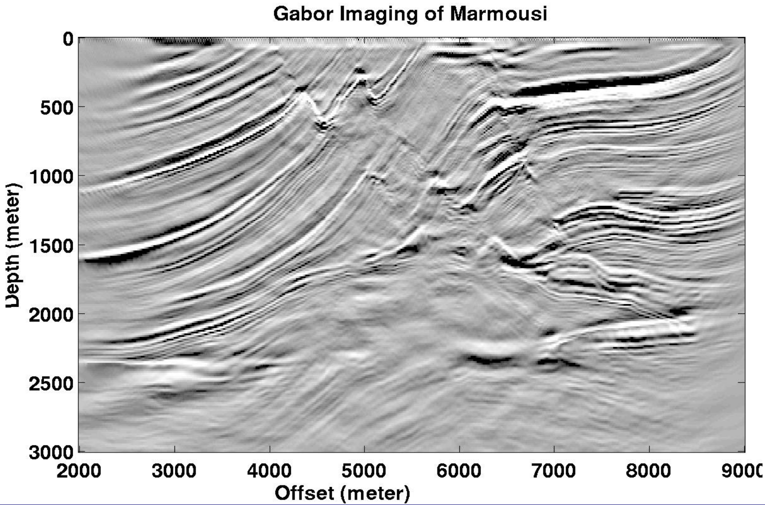
Marmousi Velocity Model



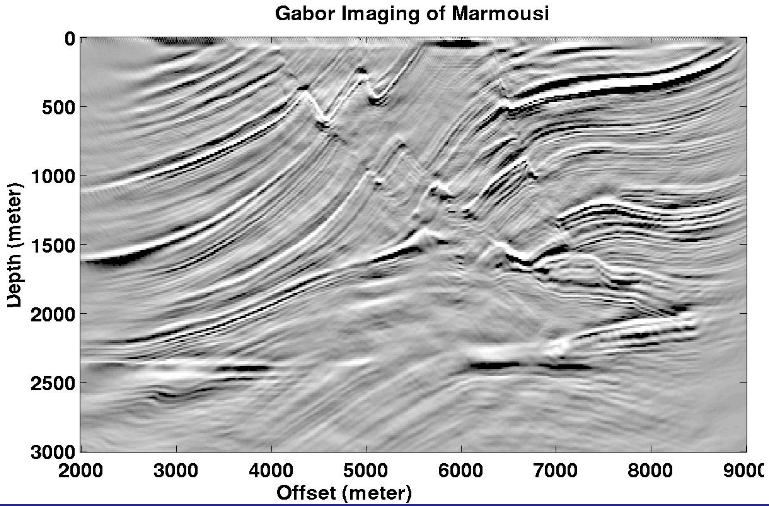
No Split-step Fourier Corrections (factor=5,22.5 hrs)



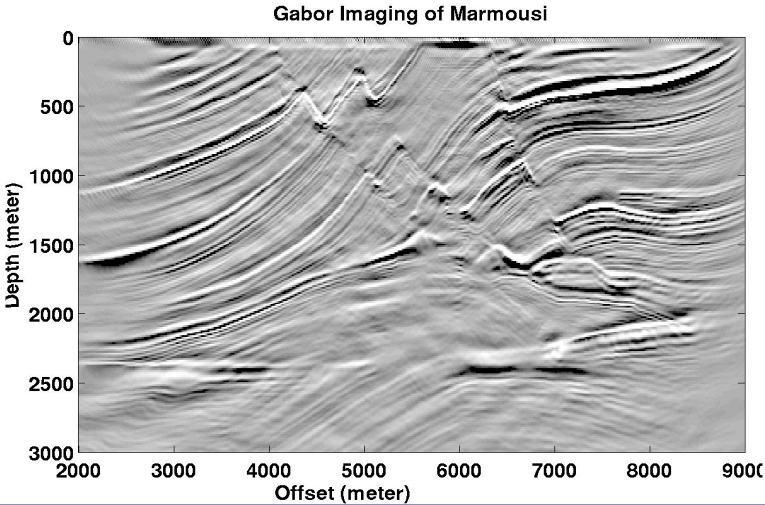
With Split-step Fourier Corrections (factor=5,22.5 hrs)



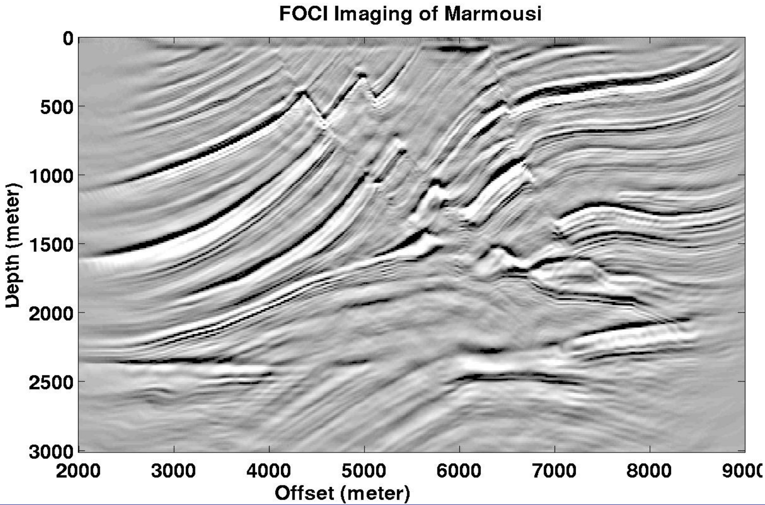
With VGAW Algorithm (factor=10, 45 hrs)



With VGAW Algorithm (factor=20,112 hrs)



FOCI (51 point, 20 hrs)



Conclusions

- Very good algorithms for accurate imaging
- Possibility of making fast imaging
- Control over imaging accuracy and speed

Future Work

- Phase error adaptive windowing algorithm - imaging with more physically-meaning criteria
- Compactly-supported windows - improving depth migration speed
- 3D complex velocity model depth imaging - giving a new depth imaging option in 3D

Acknowledgments

CREWES Sponsors

CREWES

Dave Henley Kevin Hall

Linping Dong Carlos Montaña

Chad Hogan Pavan Elapavuluri

Till Tantau (The inventor of Beamer - a LaTeX class for creating these slides)