A Velocity Smoothing Technique Based on Damped Least Squares

Zhenyue Liu

ABSTRACT

Velocity smoothing benefits some migration algorithms by stabilizing ray tracing and improving images. From the fact that a smooth velocity must have small derivatives, I develop an algorithm for velocity smoothing based on the damped least-squares (DLS) method. Unlike other convolutional-type approaches, the computational cost in the proposed smoothing method is independent of the length of the smoothing operator, so the DLS method works efficiently for applications to migration. Moreover, this method allows flexibility for changing the length of the smoothing operator at different spatial locations.

A velocity smoothing process can be viewed as an averaging process over a specific range. From this viewpoint, I define the concept of length for a general smoothing operator which sets up a standard to measure smoothing parameters of different smoothing methods.

Migration image may suffer if the velocity used in migration is oversmoothed. In order to judge if a velocity field is oversmoothed, I have derived a relationship between the migration error and the smoothing parameter for a simple case, which offers a reference for choosing the smoothing parameter. Furthermore, a perturbation technique is used to correct the traveltime distortion due to an oversmoothed velocity in the Kirchhoff integral.

Numerical examples show applications of the proposed velocity-smoothing method to Gaussian beam migration and Kirchhoff migration.

INTRODUCTION

A smooth velocity function is required by some migration algorithms. For example, the WKBJ approximation and the stationary-phase method are based on the assumption that the dominant wavelength in propagating waves must be much less than the scale of the medium variation (Bleistein, 1984). This assumption is valid only when both the medium velocity and density are smooth. Even though velocity discontinuities can be handled by boundary and interface conditions, there the interface itself must satisfy a similar smoothness condition. Velocity smoothing will benefit the solver of eikonal equation. The upwind finite-difference method (van

Trier and Symes, 1991; Pusey and Vidale, 1991) provides an efficient way to solve the eikonal equation for both traveltimes and amplitudes. However, this method will fail if turned rays exist in wave propagations. A turned ray is so defined that the ray path parallels the computational front. A smoothed velocity field can suppress the curvature of ray paths to avoid turned rays (Liu, 1993). In addition, velocity smoothing will reduce the effect of an erroneous velocity field from NMO velocity analysis. A velocity field obtained from normal moveout usually contains anomalies if velocity contains a lateral variation. These anomalies may result in a distortion of migration image. Velocity smoothing can reduce the effect of the anomalies and give a reasonable imaging map. Moreover, a smooth velocity is also helpful in reducing numerical errors in computational methods, such as interpolation and differentiation. Besides the above advantages, velocity smoothing also benefits the determination of the velocity model in velocity analysis (Versteeg, 1993) and the attenuation of unwanted multiple reflections in two-way wave equation algorithms.

There are several approaches to velocity smoothing, such as box-car smoothing, triangle averaging, and Gaussian filtering. These approaches can be represented by convolutional-type smoothing operators. The effect of a smoothing operator also depends on how much a smoothing parameter is used. Smoothing parameters in different operators may have different interpretations. To make a quantitative comparison between smoothing operators, I define the concept of length for a general smoothing operator. This concept sets up a standard to measure smoothing parameters used in different methods.

Computational cost in a convolutional-type smoothing method is proportional to the length of the smoothing operator. That is, the longer the operator, the more costly the smoothing process becomes. Another problem is that some regions in a velocity field may require more smoothing than others. Conventional smoothing methods cannot easily change the operator length to accommodate local smoothing requirements.

Here, I present a velocity-smoothing technique based on damped least squares (DLS). A smooth velocity function is sought that minimizes the weighted sum of (i) the deviation between the smooth velocity and the original one, and (ii) the first derivatives of velocity. This minimization problem is equivalent to a second-order differential equation that can be solved very efficiently. With help of an iterative feature, the DLS method can smooth higher-order derivatives in order to satisfy various purposes. Unlike other methods, computational cost in the DLS method is independent of the length of the smoothing operator. Therefore, the DLS will work efficiently if velocity smoothing needs a long smoothing operator. For example, I have found that the computational cost in the DLS is less than that in the triangle-averaging method if more than 9 points are used in the triangle-smoothing operator. Typically, geophycists use 24 or 48 points (one cable length) in the operator to smooth velocity for migration. Moreover, by using a weighting function, the DLS method can handle a local variation in the degree of velocity smoothing, so that the velocity can be made smoother in desired areas.

A proper choice of smoothing parameters is essential. An oversmoothed velocity may cause image distortions in migration processes. In this paper, I derive a formula to show the relationship between migration error and smoothing parameter used. This formula yields a criterion to judge if velocity is oversmoothed. In some applications, the smoothing parameter has to be chosen over largely in order to satisfy other criteria such as avoiding turned rays. The traveltime distortion due to an oversmoothed velocity can be corrected by using a perturbation method. For instance, in the Kirchhoff integral, I calculate raypaths and ray parameters in the smoothed velocity. Fermat's principle states that the raypath in the smoothed velocity changes little compared to that in the original velocity. This principle allows me to compute a perturbed traveltime from the velocity perturbation to correct the traveltime in the migration algorithm.

GENERAL DESCRIPTION OF THE SMOOTHING OPERATOR

Let us start with the one-dimensional case. A smoothing process can be viewed as a convolution operator. Suppose that $v(x)$ is a 1-D velocity field. The smoothed version of this velocity, $v_s(x)$, generally can be represented by

$$v_s(x) = s_\alpha * v \equiv \int_{-\infty}^{\infty} s_\alpha(x-\xi)v(\xi)d\xi, \tag{1}$$

where s_α is the kernel function of the smoothing operator that depends on a smoothing parameter, α. The smoothing parameter is used to control the degree of velocity smoothing.

Example 1. Rectangle-averaging (box-car smoothing) method. The kernel function is a rectangle, defined by

$$b_l(x) = \begin{cases} 1/l, & \text{if } |x| \leq l/2, \\ 0, & \text{if } |x| > l/2, \end{cases} \tag{2}$$

where the smoothing parameter l is the width of the rectangle. Note that this kernel function has jumps at $x = \pm l/2$.

Example 2. Triangle-averaging method. The kernel function is a triangle, defined by

$$s_\alpha(x) = \begin{cases} 2(1-2|x|/\alpha)/\alpha, & \text{if } |x| \leq \alpha/2, \\ 0, & \text{if } |x| > \alpha/2, \end{cases} \tag{3}$$

where the smoothing parameter α is the length of the triangle base. This kernel function is continuous but its derivative has jumps with magnitude $8/\alpha^2$ at $x = 0$.

Example 3. Gaussian filtering method. The kernel function is

$$s_\alpha(x) = \frac{1}{\sqrt{2\pi}\alpha} \exp(-\frac{x^2}{2\alpha^2}). \tag{4}$$

This kernel function is infinitely differentiable.

The kernel of a smoothing operator should be characterized by the following properties:

(1) Normality,

$$\int_{-\infty}^{\infty} s_\alpha(x)dx = 1. \tag{5}$$

Equation (5) means that a smoothing operator does not change a constant velocity. In fact, if $v(x)$ is a constant v, then

$$[s_\alpha * v](x) = \int_{-\infty}^{\infty} s_\alpha(x-\xi)v d\xi = v\int_{-\infty}^{\infty} s_\alpha(x-\xi)d\xi = v. \tag{6}$$

(2) Nonnegativity,

$$s_\alpha(x) \geq 0 \tag{7}$$

Under this condition,

$$\begin{aligned} \min_x\{[s_\alpha * v](x)\} &= \min_x \int_{-\infty}^{\infty} s_\alpha(x-\xi)v(\xi)d\xi \\ &= \min_x \int_{-\infty}^{\infty} s_\alpha(\xi)v(x-\xi)d\xi \\ &\geq \int_{-\infty}^{\infty} s_\alpha(\xi)\min_x v(x-\xi)d\xi \\ &= \min_x v(x)\int_{-\infty}^{\infty} s_\alpha(\xi)d\xi \\ &= \min_x v(x). \end{aligned} \tag{8}$$

That is,

$$\min_x [s_\alpha * v](x) \geq \min_x v(x). \tag{9}$$

In the same way, one can prove that

$$\max_x [s_\alpha * v](x) \leq \max_x v(x). \tag{10}$$

Inequalities (9) and (10) imply that when a smoothing operator acts on a step function, it should not produce the Gibbs phenomemon as a low-pass filter does.

(3) Monotonicity. For each α, $s_\alpha(x)$ decreases with an increase of $|x|$.

(4) Spectral monotonic convergence; i.e., if $S_\alpha(k)$ is the Fourier transform of $s_\alpha(x)$, then

$$\lim_{\alpha\to\infty} S_\alpha(k) = 0, \text{ for } k \neq 0, \tag{11}$$

monotonically. That is, the limit of a smoothed velocity is a constant.

Although the smoothing operator defined in equation (1) is one dimensional, it will work for both the 2-D and the 3-D cases. In fact, In higher dimensions, a velocity can be smoothed recursively in each dimension; then the final output velocity will be smooth in the whole domain in all variables.

DAMPED LEAST-SQUARES METHOD

Given the kernel function in equation (2), (3), or (4), the smoothed velocity can be calculated directly from the convolutional integral. The computational cost is proportional to the smoothing parameter used, and, therefore, is intensive when a large smoothing parameter is required. In contrast, velocity smoothing based on a differential equation may solve this problem. Mathematically, a function is smooth if it has small derivatives. From this viewpoint, I use the damped least-squares method to smooth velocity via suppressing its derivatives.

Variational and Differential Forms

In the damped least-squares method, the smoothed velocity $v_s(x)$ is determined by

$$\int w(x)\,(v_s(x)-v(x))^2 dx + \alpha^2 \int \left(\frac{dv_s}{dx}\right)^2 dx = \min, \tag{12}$$

where α is the smoothing parameter and $w(x)$, ranging from 0 to 1, is a weighting function. The equivalent differential form of minimization (12) is

$$w(x)v_s(x) - \alpha^2 \frac{d^2 v_s}{dx^2} = w(x)v(x). \tag{13}$$

In discretization, the above differential equation becomes a tridiagonal linear system that can be solved very efficiently.

The minimization (12) smooths a velocity by suppressing its first derivative. Generally, the smoothing effect will be better if higher-order derivatives are suppressed. One could add the quadratic terms of higher-order derivatives of v_s into the minimization (12). However, the higher derivatives will result in a discrete linear system with more than three diagonals, which is more difficult to solve. Moreover, as I will show later, the kernel function of the smoothing operator generated in this way may not be nonnegative, which violates the requirement (7). In order to suppress the higher-order derivatives, here, I use an iterative scheme in equation (13); i.e.,

$$w(x)v_n(x) - \alpha^2 \frac{d^2 v_n}{dx^2} = w(x)v_{n-1}(x),\ n = 1, 2, ... \tag{14}$$

Here $v_n(x)$ is derived from v_{n-1} and I assume that $v_0(x) = v(x)$. The equivalent variational form of the above equation, for $n = 2$, is

$$\int w^2 (v_2 - v)^2 dx + 2\alpha^2 \int w \left(\frac{dv_2}{dx}\right)^2 dx + \alpha^4 \int \left(\frac{d^2 v_2}{dx^2}\right)^2 dx = \min; \tag{15}$$

i.e., the derivatives up to the second order of $v_2(x)$ are suppresed. Thus, inductively, one can conclude that equation (14) has suppressed the higher-order derivatives of the smoothed velocity for $n \geq 2$. I call the smoothing operator defined in equation (14) the *nth-order DLS* smoothing operator if n iteration steps are taken in equation (14).

Kernel Function of DLS

By following equation (14) and setting $w(x) \equiv 1$, the smoothed velocity in the wavenumber domain is represented by

$$V_n(k) = \frac{V_{n-1}(k)}{1+\alpha^2 k^2}, \tag{16}$$

where V_n and V_{n-1} are the Fourier transforms of v_n and v_{n-1} respectively. Repeating equation (14) n times, I have

$$V_n(k) = V(k)\,(1+\alpha^2 k^2)^{-n}, \tag{17}$$

where V is the Fourier transform of v. Therefore, large-wavenumber components in the smoothed velocity v_n are suppressed.

Now, the explicit representation of v_n can be calculated by using equation (17). Taking an inverse Fourier transform on equation (17) yields

$$v_n(x) = s_{n,\alpha} * v = \int_{-\infty}^{\infty} s_{n,\alpha}(x-\xi)v(\xi)d\xi, \tag{18}$$

where $s_{n,\alpha}$ is the inverse Fourier transform of $(1+\alpha^2k^2)^{-n}$. From the table of intergral transforms, one can find out that

$$s_{n,\alpha}(x) = \frac{(-1)^{n-1}\sigma^{2n-1}}{(n-1)!2^n\alpha}\left(\frac{d}{\sigma d\sigma}\right)^{n-1}\frac{e^{-\sigma}}{\sigma}\Bigg|_{\sigma=\frac{|x|}{\alpha}}. \tag{19}$$

Suppressing derivatives up to the second order or the third order suffices in most applications. The first three functions defined in equation (19) are

$$s_{1,\alpha}(x) = \frac{1}{2\alpha}e^{-\frac{|x|}{\alpha}}, \tag{20}$$

$$s_{2,\alpha}(x) = \frac{1}{4\alpha}\left(1+\frac{|x|}{\alpha}\right)e^{-\frac{|x|}{\alpha}}, \tag{21}$$

and

$$s_{3,\alpha}(x) = \frac{1}{16\alpha}\left(3+3\frac{|x|}{\alpha}+\frac{x^2}{\alpha^2}\right)e^{-\frac{|x|}{\alpha}}. \tag{22}$$

These three kernel functions are nonnegative. In fact, the nonnegativity is true for all kernel functions in equation (19). This conclusion follows from

$$s_{n,\alpha}(x) = s_{n-1,\alpha}(x) * s_{1,\alpha}(x). \tag{23}$$

Notice that not all kernel functions derived by damped least-squares methods are nonnegative. For example, if a smoothed velocity is determined by only suppressing its second derivative, i.e,

$$\int (v_s(x)-v(x))^2dx + \frac{\alpha^4}{4}\int\left(\frac{d^2v_s}{dx^2}\right)^2 dx = \min, \tag{24}$$

then one can prove that the kernel function of this smoothing operator is

$$s_\alpha(x) = \frac{1}{2\alpha}[\sin(|x|/\alpha) + \cos(|x|/\alpha)]e^{-\frac{|x|}{\alpha}}. \tag{25}$$

This function has a value $-e^{-\pi}/2\alpha$ at $x = \alpha\pi$. Therefore, suppressing only the second derivative is not sufficient for velocity smoothing.

Although equation (18) gives an analytical representation for the smoothed velocity in the DLS Method, the computation cost involved in this integral is proportional to $O(N^2)$. Here N is the number of velocity samples. In comparison, the computation cost involved equation (14) is only proportional to $O(N)$. Therefore, it is preferable to solve equation (14) for the smoothed velocity in the DLS method.

Compared to the DLS, computational cost in other convolutional-type smoothing methods is proportional to $O(N_l N)$. Here N_l is the number of samples in the smoothing operator. Therefore, computational cost in the DLS method will be less than that in convolutional-type smoothing if many samples in smoothing operators are used. For example, I have found that the computational cost in the DLS is less than that in the triangle-averaging method if more than 9 points are used in the smoothing operator. In fact, geophysicists often use 24 or 48 points (one cable length) for smoothing velocity.

A Choice of Weighting Function

The weighting function $w(x)$ in equation (12) imposes a locality constraint between the original velocity and the smoothed one. A smaller value of the weighting function at some point shows that the difference between the original velocity and the smoothed one is larger at this point or that the velocity near this point is smoothed more than elsewhere. From equation (13), the smoothing parameter α approximately increases by $1/w(x)$ compared to the case, $w(x) \equiv 1$. A useful choice is that the weighting function decreases with depth, to account for the deteriorating accuracy of the velocity model with depth if this model is obtained from velocity analysis (Liu and Bleistein, 1991). For example, one can choose

$$w(x) = \frac{1}{1 + \mu^2 x^2}, \tag{26}$$

where x is the depth and μ is a constant. The larger value μ, the faster the function $w(x)$ decreases with depth.

LENGTH OF SMOOTHING OPERATOR

The degree of a smoothing operator depends on the value of its smoothing parameter. However, definitions of smoothing parameters in different operators may be inconsistent for each other. Therefore, it is desirable to set up a standard to measure the smoothing degree for a general smoothing operator.

A smoothing operator actually performs an averaging process. For example, the rectangle-averaging method in equation (2) averages the original velocity over the rectangle width. The greater the width of the rectangle, the smoother the output velocity will be. From this viewpoint, the rectangle width that characterizes the smoothing degree of the rectangle-averaging operator can be used to define the length of this smoothing operator. Here, I generalize this concept to other smoothing operators. Given a smoothing operator s_α, for each smoothing parameter α, there exists a width of rectangle, l, such that

$$\int_{-\infty}^{\infty} |s_\alpha(x) - b_l(x)| dx = \min . \tag{27}$$

I define this $l = l(\alpha)$ as the length of s_α at the smoothing parameter α. In other words, one can say that the rectangle operator with length (width) $l(\alpha)$ is equivalent to the smoothing operator s_α. In Appendix A, I prove that the minimization problem (27) is equivalent to a nonlinear equation

$$x_l = \frac{l^2}{2} s_\alpha(l/2), \tag{28}$$

where $0 < x_l < l/2$ and is the solution of

$$\frac{1}{l} = s_\alpha(x). \tag{29}$$

If the kernel function furthermore satisfies the homogeneity condition

$$s_\alpha(x) \equiv \frac{1}{\alpha} s_1(x/\alpha), \tag{30}$$

then l can be represented by a linear function of α; i.e., there exist a contant q such that

$$l = q\alpha. \tag{31}$$

Note that all of the kernel functions in the examples (3), (4), and in the DLS method satisfy the condition (30). By using the linear relationship (31), equations (28) and (29) can be rewritten as

$$q_l = \frac{q^2}{2} s_1(q/2), \tag{32}$$

$$\frac{1}{q} = s_1(q_l), \tag{33}$$

where q_l is an auxillary variable.

For the triangle-averaging method, equations (32) and (33) become

$$q_l = q^2(1 - q), \tag{34}$$

$$\frac{1}{q} = 2 - 4q_l; \tag{35}$$

hence,

$$q = \frac{1}{\sqrt{2}}, \tag{36}$$

and

$$l = \frac{\alpha}{\sqrt{2}} = 0.707\,\alpha, \tag{37}$$

For the Gaussian-filtering method, equations (32) and (33) become

$$q_l = \frac{q^2}{2\sqrt{2\pi}} \exp(-\frac{q^2}{8}), \tag{38}$$

$$\frac{1}{q} = \frac{1}{\sqrt{2\pi}} \exp(-\frac{q_l^2}{2}), \tag{39}$$

with solution

$$q = 2.973. \tag{40}$$

and

$$l = 2.973\,\alpha. \tag{41}$$

For the damped least-squares methods determined by equations (20) to (22), I obtain

$$q = 3.389,\ 5.187,\ 6.595. \tag{42}$$

Using the length of a smoothing operator allows us to choose a smoothing parameter according to the range over which the original velocity is required to be averaged. Because triangle averaging is a widely-used smoothing method, it is preferable to use the length of the triangle base in the triangle operator to measure the smoothing parameter for other operators. From equation (36), the length of the triangle base in the triangle operator, L, is the length of the equivalent rectangle operator multiplied by $\sqrt{2}$, i.e., $L = l\sqrt{2}$. Therefore, q-values of smoothing operators in equations (40) and (42) should be multiplied by $\sqrt{2}$, if the smoothing parameters are measured based on the length of the triangle base, L. For example, the q-values in equations (42) will become

$$q = 4.793,\ 7.336,\ 9.327. \tag{43}$$

Figure 1 shows the three kernel functions of the damped least squares (DLS) that have the same operator length $L = \sqrt{2}$. The function for $n = 1$ has a sharp peak at $x = 0$, so this operator cannot smooth a velocity very effectively. The functions for $n = 2$ and $n = 3$ are horizontal at $x = 0$, so these operators will work better. In fact, the difference between the cases $n = 2$ and $n = 3$ is not significant. Figure 2 shows comparison among the DLS smoothing operator for $n = 3$, triangle-averaging and Gaussian-filtering operators. All of them have the same operator length $L = \sqrt{2}$. The Gaussian operator has the smoothest kernel function and the triangle operator is the least smooth. On the other hand, a smoothing process via the Gaussian operator is more costly than the triangle operator, because the Gaussian operator involves an

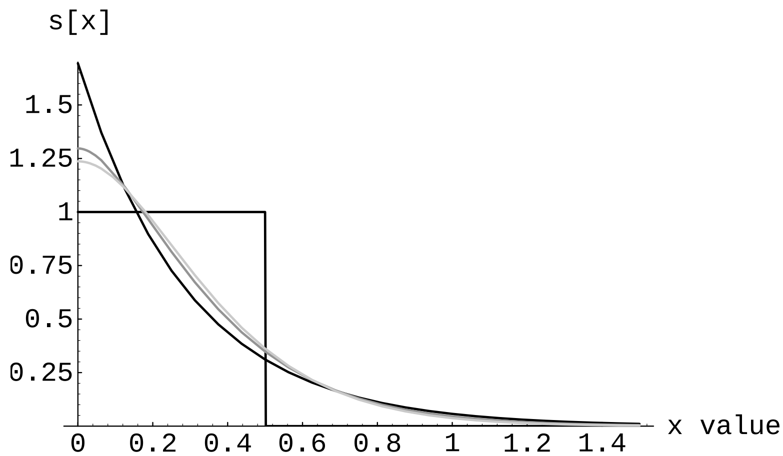

FIG. 1. The positive half of the DLS kernel functions. The black curve denotes the case $n = 1$; the lightest-gray, the case $n = 3$; the medium gray, the case $n = 2$.

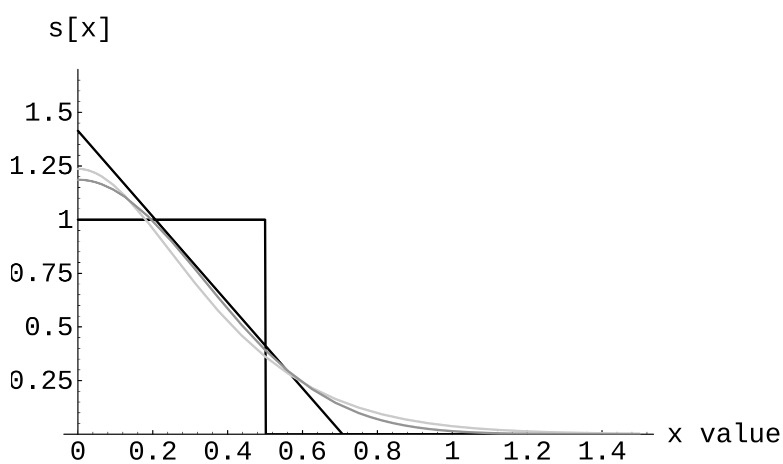

FIG. 2. The positive half of kernel functions. The black curve denotes the triangle operator; the lightest-gray, the DLS operator; the medium gray, the Gaussian operator.

exponential function and has more nonzero function samples. Compared to these two methods, computational cost in the DLS method that solves a differential equation is independent of the sampling number and, therefore, is efficiently when a large length of smoothing operator is required. For example, computational cost in the DLS method for $n = 2$ will be less than that in the triangle averaging method if the sampling number is more than 9.

APPLICATION TO MIGRATION

Choice of a smoothing parameter

A suitable choice of a smoothing parameter is required for the velocity-smoothing process. An undersmoothed velocity cannot effectively eliminate migration artifacts arising from velocity discontinuities. On the other hand, an oversmoothed velocity may result in image distortion due to traveltime error. Therefore, it is essential to study the relationship between migration error and smoothing parameters used. Here, I derive an analytical formula to describe this relationship for a simple model.

Consider the one dimensional case. Suppose that a slowness distribution is defined by a piecewise constant function,

$$p(z) = \begin{cases} p_1, & \text{if } 0 < z < d, \\ p_2, & \text{if } z \geq d. \end{cases} \tag{44}$$

Suppose that this slowness is smoothed via the rectangle-averaging with the width, l. In Appendix B, I prove that the traveltime error at at $z = d$ is

$$\Delta t = \frac{1}{8}(p_2 - p_1)l, \tag{45}$$

and that the migration depth error at $z = d$ is

$$\Delta z = \frac{l\,(p_2 - p_1)}{4(p_1 + p_2)}. \tag{46}$$

If the dominant wavelength in the propagation wave is λ, then, in order to avoid the imaged depth distortion, one should have $|\Delta z| \leq \lambda/4$. From equation (46), l will satisfy

$$l \leq \frac{(p_1 + p_2)\lambda}{|p_2 - p_1|}. \tag{47}$$

If L is the length of the triangle-averaging operator, then, from equation (36), the relationship (47) will become

$$L \leq \frac{\sqrt{2}(p_1 + p_2)\lambda}{|p_2 - p_1|}. \tag{48}$$

This inequality yields a criterion for choosing the smoothing parameter that depends on the propagating wavelength and the magnitude of the velocity discontinuity as well. For example, if the ratio p_1/p_2 in a velocity field is 200%, the relationship (48) implies $L \leq 3\sqrt{2}\lambda \approx 4\lambda$. That is, the smoothing parameter should not be more than four wavelengths to avoid image distortation.

Gaussian Beam Migration

In Gaussian beam migration, if the algorithm does not handle interfaces in the background velocity, the grid velocity will be interpolated as a continuous function so that one can calculate ray parameters. In order to reduce errors in numerical computation, this interpolation certainly needs a smooth velocity model. To illustrate the effect of velocity smoothing, I compare Gaussian-beam migration results with velocity smoothing and without it. The algorithm of Gaussian beam migration (Hale, 1992) used here cannot handle interfaces in the background velocity.

Figure 3 shows a constant-layered velocity model that has strong discontinuities. The maximum contrast in the velocity (between the first and the third layer) is about 100%. By using the ray-tracing method, I generate the zero-offset synthetic data from this model, shown in Figure 4.

Figure 5 shows the data migrated with the original velocity. The migration result is poor: low resolution and migration artifacts appear in all reflectors and the fourth reflector is almost invisible. Figure 6 shows a velocity model smoothed via the second-order DLS method. The length of the smoothing operator is 320 meters in the horizontal direction and 160 meters in the vertical direction. Figure 7 shows the Gaussian beam migration using this smoothed velocity. This result is much better than that in Figure 6 and quite acceptable. Figure 8 shows the Gaussian beam migration using the velocity smoothed via the third-order DLS method. Differences between Figure 7 and Figure 8 are imperceptible. In fact, the difference between the smoothed velocities via these two approaches is only in the third derivative of the velocity function which affects little on migration result. This example suggests that the second order DLS method suffices in applications to migration. Figure 9 shows the Gaussian beam migration using the velocity smoothed via the rectangle averaging method. The length of smoothing operator is the same as that of the DLS method. This result is good as well, but looks worse than Figure 7. One can see many wiggles between the second interface and third one. Therefore, the DLS method produces a smoother velocity model than rectangle averaging, if they use the same operator length. Figure 10 shows the Gaussian beam migration using the velocity smoothed via the triangle averaging method. This result is as good as Figure 7. Since the sampling numbers are 13 both in the horizontal direction and in the vertical direction, velocity smoothing via the triangular averaging method takes about 50% computational time more than the second-order DLS method.

Kirchhoff migration

Ray parameters in the Kirchhoff integral are calculated either by ray tracing or finite difference. Although a ray tracing algorithm can be designed to handle velocity interfaces, interface smoothing is required for obtaining a stable ray tracing algorithm (Liu and Bleistein, 1991). In the finite-difference approach, van Trier and Symes (1991) gave an efficient upwind algorithm to calculate traveltimes and traveltime derivatives. However, this method needs a sufficiently smooth velocity to avoid turned rays for a successful implementation.

If the original velocity has very strong discontinuities, the velocity for the upwind algorithm has to be smoothed too much to satisfy the criterion (48); i.e, the velocity is oversmoothed. An oversmoothed velocity will result in traveltime errors and cause image distortion in migration. Here I use a perturbation technique to correct this image distortion. I calculate the ray parameters (raypath) in the smoothed velocity. From Fermat's principle, this raypath will be approximately the same as that in the original velocity if the velocity perturbation between these two velocities is sufficiently small. Then I correct the traveltime from the velocity perturbation based on the raypath calculated in the smoothed velocity. By using the corrected traveltimes, one will obtain a more precise migration image. The detailed formulas are derived in Appendix C.

To illustrate the perturbation technique, the finite-difference Kirchhoff migration is implemented using a velocity model with strong discontinuities shown in Figure 11. The maximum contrast in the velocity is 1 to 3.2 at the location, $x = 0$ and $z = 1.75$ km. A common-offset synthetic section is generated with this velocity model by using Gaussian beam modeling (Rüger, 1993). The velocity is smoothed via the second-order DLS method. For a successful implementation of the upwind finite-difference method, I have to smooth the velocity by using very large operator lengths: 2.5 km in the vertical direction and 0.5 km in the horizontal direction. The smoothed velocity shown in Figure 13 has the wavelengths approximately 5 km in the vertical direction and 6 km in the horizontal direction. The migration result with this velocity shows that this velocity is obviously oversmoothed (Figure 14). Compared to the true model in Figure 11 one can see that the lense is squeezed and that the bottom reflector deviates from the flat position. After using the perturbation technique, the migration output provides a correct structural image, shown in Figure 15. The perturbation only costs about 11% additional computation.

CONCLUSION

Velocity smoothing is helpful in reducing the migration error due to the high frequency assumption and interpolation calculations. Here, I have discussed the properties of smoothing operators and derive the DLS method to smooth a velocity by suppressing velocity derivatives. Compared to convolutional-type approaches, the computational cost in the DLS method is independent of the length of smoothing operator, so the DLS method works efficiently for applications to migration. Moreover, this method allows flexibility for changing the length of the smoothing operator at different spatial locations.

A suitable smoothing parameter should be chosen according to the propagating wavelength and the magnitude of the velocity discontinuity. In some applications, an overlarge smoothing parameter has to be used for a stable computation of traveltimes, for example; the traveltime errors due to the oversmoothed velocity should not be

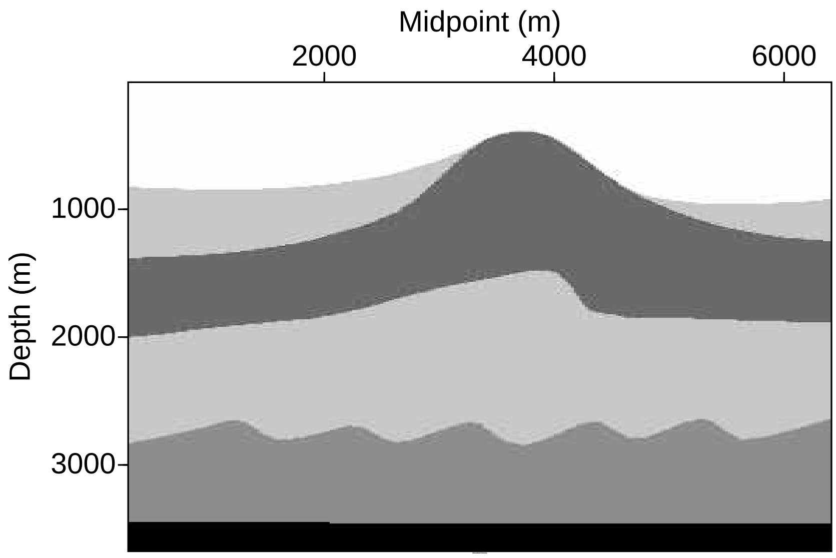

FIG. 3. The original velocity model in the test of Gaussian beam migration. The darker shading denotes higher velocity.

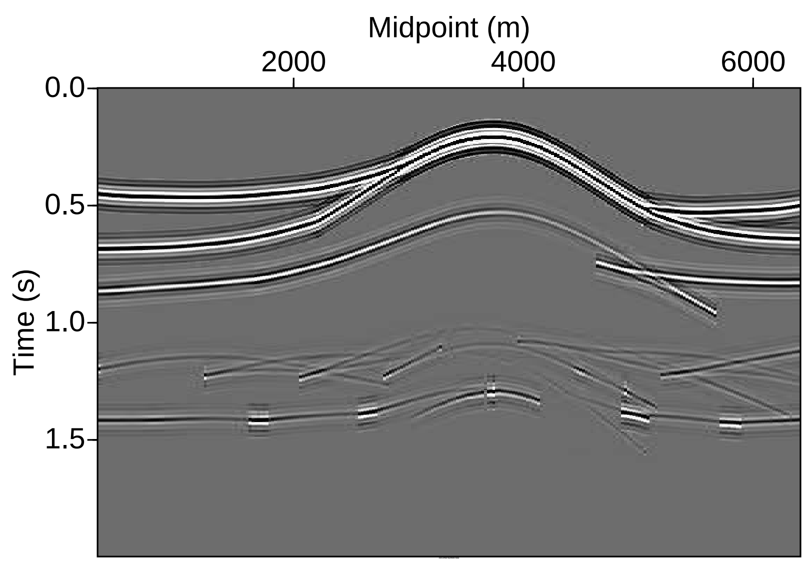

FIG. 4. Zero-offset synthetic data generated with the velocity model in Figure 3.

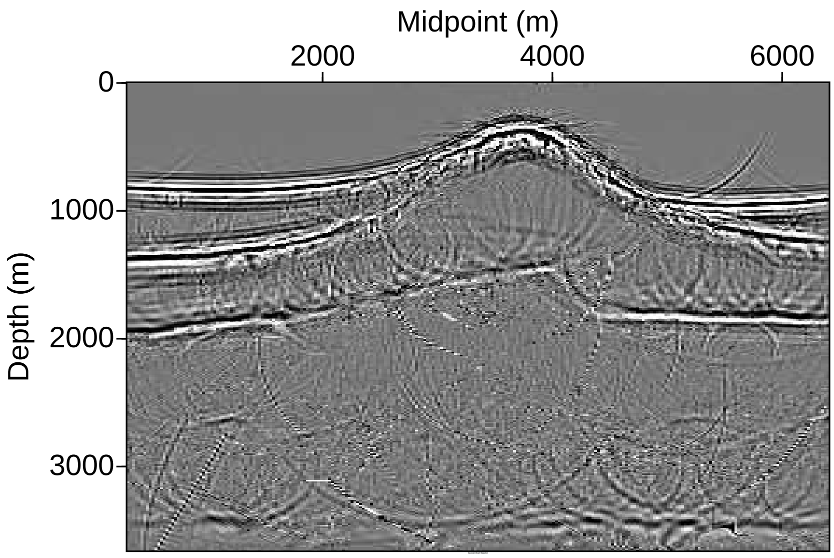

FIG. 5. Gaussian beam migration with the original velocity model.

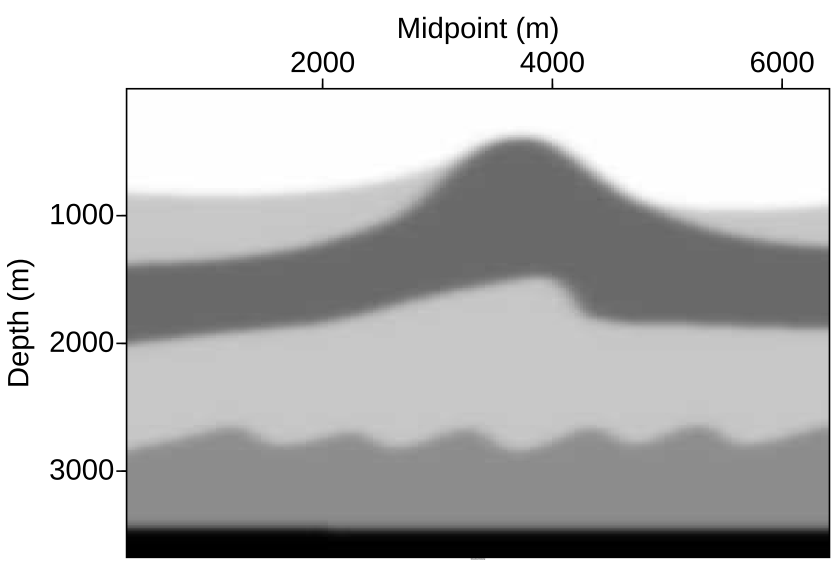

FIG. 6. The velocity model smoothed by the second-order DLS method. The darker shading denotes higher velocity.

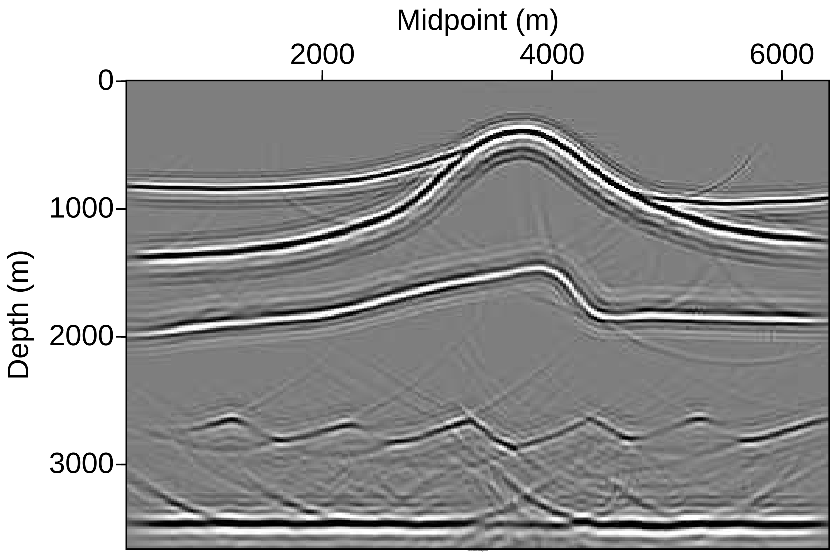

FIG. 7. Gaussian beam migration with the smoothed velocity in Figure 6.

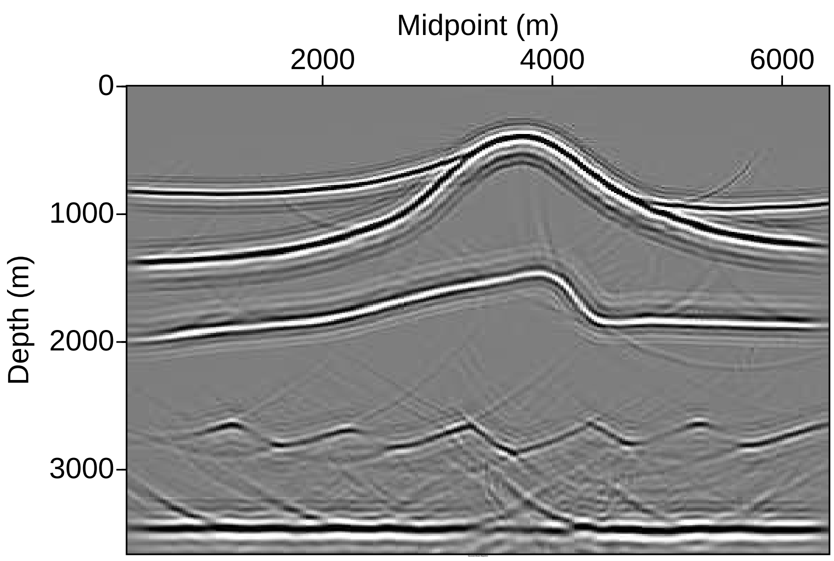

FIG. 8. Gaussian beam migration with the smoothed velocity via the third-order DLS method.

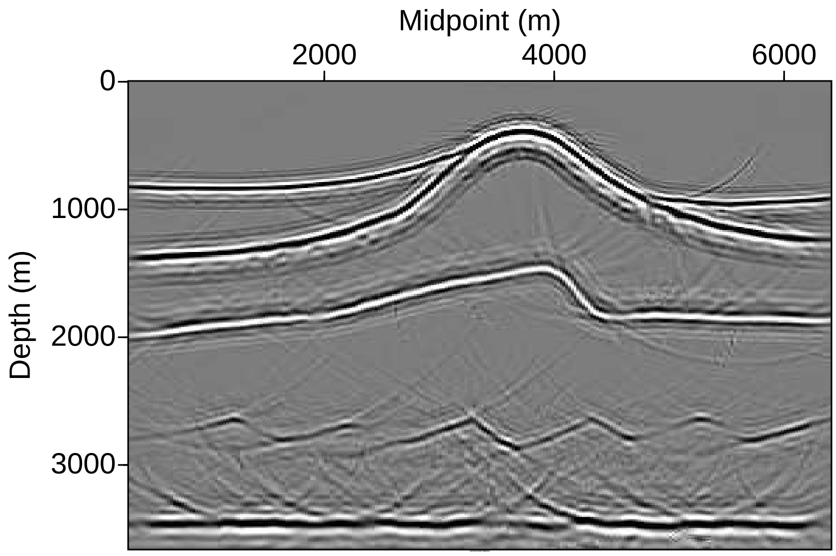

FIG. 9. Gaussian beam migration with the velocity smoothed via rectangle averaging.

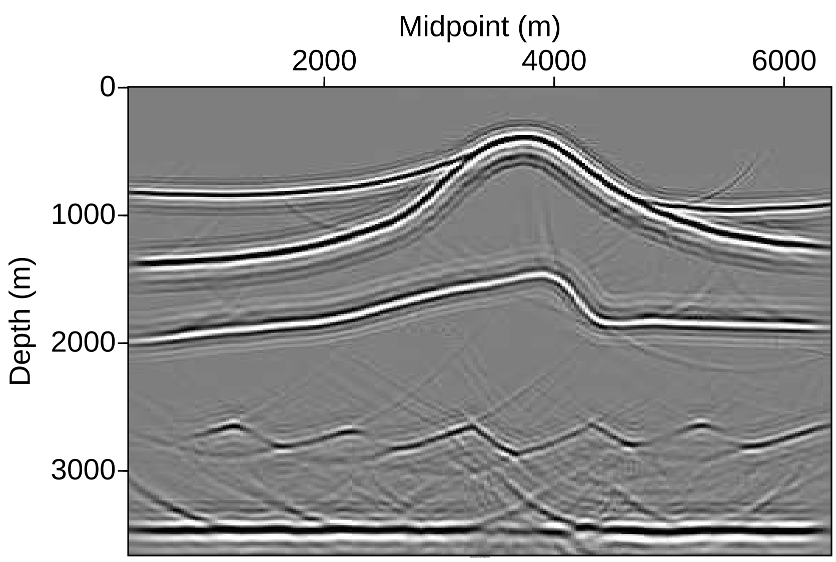

FIG. 10. Gaussian beam migration with the velocity smoothed via triangle averaging.

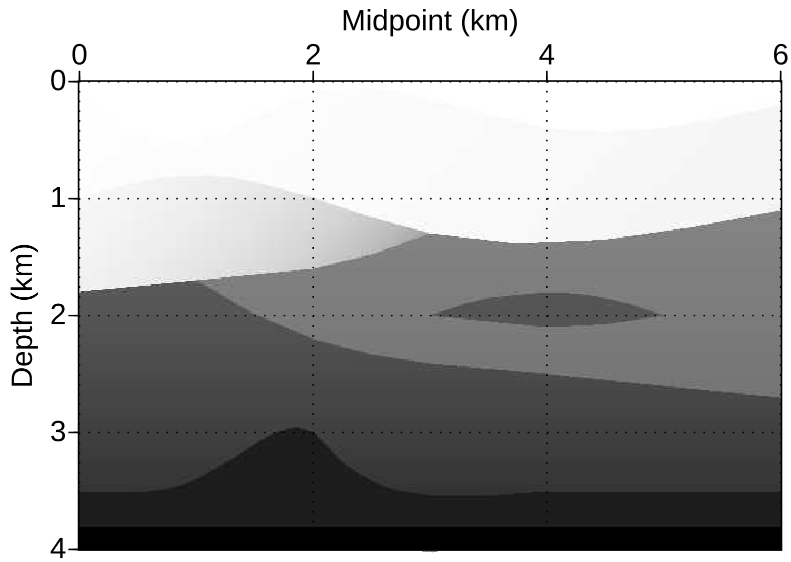

FIG. 11. The original velocity model in the test of Kirchhoff migartion. The darker shading denotes higher velocity.

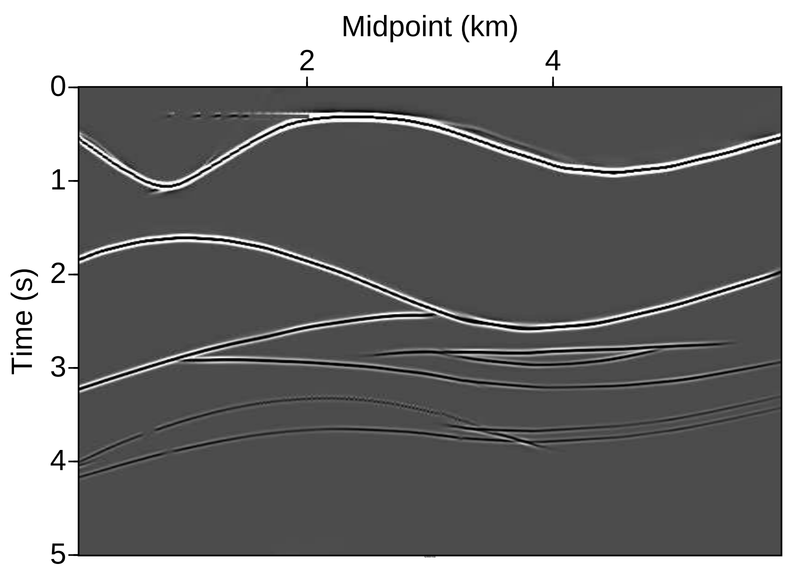

FIG. 12. Common-offset synthetic data generated with the velocity model in Figure 11. The offset is 0.3 km.

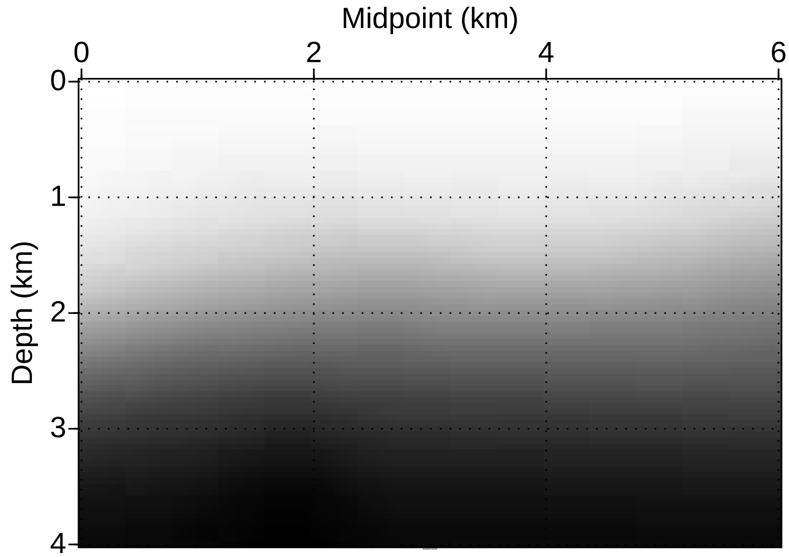

FIG. 13. The smoothed velocity model. The darker shading denotes higher velocity.

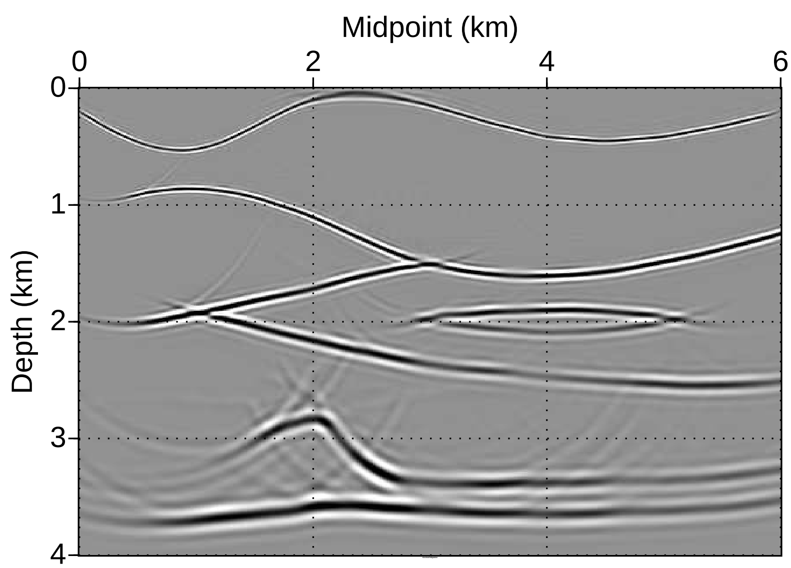

FIG. 14. Kirchhoff migration with the velocity in Figure 13. Traveltime pertubation is not used.

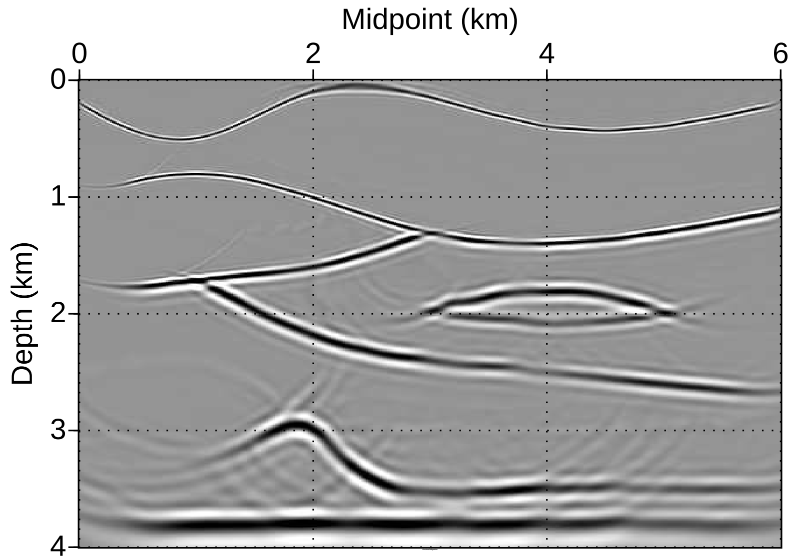

FIG. 15. Kirchhoff migration with the velocity in Figure 13. Traveltime pertubation is used.

neglected. The perturbation technique can be applied to reduce these traveltime errors. In addition, smoothing methods here will work on other quantities such as density distribution or velocity interfaces.

ACKNOWLEDGMENTS

Thanks to Dr. Norman Bleistein for his help and guidance on this paper. Thanks to Dr. Zhiming Li and Dr. Robert Mager for their support and suggestions to this project during my visit to Unocal last summer. This paper was partially supported by the Office of Naval Research, Mathematics Division.

REFERENCES

Bleistein, N., 1984, Mathematical Methods for Wave Phenomena: Academic Press Inc.

Hale, D., 1992, Computational aspects of Gaussian beam migration: 1992 CWP Project Review.

Liu, Z., 1993, A Kirchhoff approach to seismic modeling and prestack depth migration: 1993 SEG expanded abstracts, p. 888.

Liu, Z., and Bleistein, N., 1991, Velocity analysis by inversion: 1991 SEG expanded abstracts, p. 1230.

Pusey, L., and Vidale, J., 1991, Accurate-difference calculation of WKBJ traveltimes and amplitudes: 1991 SEG expanded abstracts, p. 1513.

Rüger, A., 1993, Dynamic ray tracing and its application: **CWP-139**, Colorado School of Mines.

van Trier, J., and Symes, W. W., 1991, Upwind finite-difference calculation of traveltimes: Geophysics, **56**, 812-821.

Versteeg, R. J., 1993, Sensitivity of prestack depth migration to the velocity model: Geophysics, **58**, 873-882.

APPENDIX A: LENGTH OF A SMOOTHING OPERATOR

Let

$$A(l) = \int_{-\infty}^{\infty} |s_\alpha(x) - b_l(x)| dx, \tag{A-1}$$

then the minima of the function $A(l)$ will satisfy

$$\frac{\partial A}{\partial l} = 0. \tag{A-2}$$

By using the symmetry of the intergrand in equation (A-1) and the definition (2), one can rewrite equation (A-1) as

$$A(l) = 2\int_{0}^{\infty} |s_\alpha(x) - b_l(x)| dx = 2\int_{0}^{\frac{l}{2}} |s_\alpha(x) - \frac{1}{l}| dx + 2\int_{\frac{l}{2}}^{\infty} s_\alpha(x) dx. \tag{A-3}$$

Define a variable x_l such that

$$s_\alpha(x_l) = \frac{1}{l}; \tag{A-4}$$

then

$$A(l) = 2\int_{0}^{x_l} (s_\alpha(x) - \frac{1}{l}) dx - 2\int_{x_l}^{\frac{l}{2}} (s_\alpha(x) - \frac{1}{l}) dx + 2\int_{\frac{l}{2}}^{\infty} s_\alpha(x) dx. \tag{A-5}$$

Introduce the functions

$$A_1(l) = 2\int_{0}^{x_l} (s_\alpha(x) - \frac{1}{l}) dx, \tag{A-6}$$

$$A_2(l) = 2\int_{x_l}^{\frac{l}{2}} (\frac{1}{l} - s_\alpha(x)) dx, \tag{A-7}$$

and

$$A_3(l) = 2\int_{\frac{l}{2}}^{\infty} s_\alpha(x) dx; \tag{A-8}$$

then

$$A(l) = A_1(l) + A_2(l) + A_3(l). \tag{A-9}$$

Diffrentiating the above equation with respect to l and using (A-4) yields

$$\frac{\partial A_1}{\partial l} = 2\int_{0}^{x_l} \frac{1}{l^2} dx + 2\frac{\partial x_l}{\partial l}(s_\alpha(x_l) - \frac{1}{l}) = \frac{2x_l}{l^2}, \tag{A-10}$$

$$\frac{\partial A_2}{\partial l} = \frac{1}{l} - s_\alpha(l/2) + 2\frac{\partial x_l}{\partial l}(s_\alpha(x_l) - \frac{1}{l}) - 2\int_{x_l}^{\frac{l}{2}} \frac{1}{l^2}dx = \frac{2x_l}{l^2} - s_\alpha(l/2), \qquad \text{(A-11)}$$

and

$$\frac{\partial A_3}{\partial l} = -s_\alpha(l/2). \qquad \text{(A-12)}$$

Summing equations (A-10), (A-11) and (A-12), and using (A-5), one obtains

$$\frac{\partial A}{\partial l} = \frac{4x_l}{l^2} - 2s_\alpha(l/2). \qquad \text{(A-13)}$$

Substituting the about result into equation (A-2) yields

$$\frac{4x_l}{l^2} - 2s_\alpha(l/2) = 0, \qquad \text{(A-14)}$$

i.e.,

$$x_l = \frac{l^2}{2}s_\alpha(l/2). \qquad \text{(A-15)}$$

Equations (A-4) and (A-15) provides a means of obtaining l for a general smoothing operator.

APPENDIX B: MIGRATION ERROR

A smoothed slowness function via rectangle-averaging can be represented by

$$p_l(z) = \int_0^\infty p(\xi)b_l(z-\xi)d\xi, \qquad \text{(B-1)}$$

where b_l is defined by equation (2). For the slowness defined in equation (44), it is easy to find

$$p_l(z) = \begin{cases} p_1, & \text{if } 0 < z \le d - l/2, \\ p_1(d - z + l/2)/l + p_2(z + l/2 - d)/l, & \text{if } d - l/2 < z < d + l/2, \\ p_2, & \text{if } z \ge d + l/2. \end{cases} \qquad \text{(B-2)}$$

Therefore, the traveltime error at depth d between the true slowness and the smoothed one is calculated by

$$\Delta t = \int_0^d (p_l(z) - p(z))dz = \int_{d-l/2}^d (p_2 - p_1)\frac{z + l/2 - d}{l}dz = \frac{1}{8}(p_2 - p_1)l. \qquad \text{(B-3)}$$

The corresponding migration depth error at d will be

$$\Delta z = \frac{\Delta t}{p_l(d)} = \frac{(p_2 - p_1)l}{4(p_1 + p_2)}, \qquad \text{(B-4)}$$

where I use $p_l(d) = (p_1 + p_2)/2$. This verifies the claims in equations (45) and (46).

APPENDIX C: TRAVELTIME PERTURBATION

Traveltimes in the Kirchhoff intergral can be solved by using eikonal equation

$$\left(\frac{\partial \tau}{\partial x}\right)^2 + \left(\frac{\partial \tau}{\partial z}\right)^2 = p^2(x,z), \tag{C-1}$$

where p is the slowness. Suppose that

$$p(x,z) = p_s(x,z) + \delta p(x,z), \tag{C-2}$$

where p_s is the smoothed slowness and δp the slowness perturbation. From equation (C-1), then the correspoding traveltime perturbation, $\delta\tau$, will satisfy

$$\frac{\partial \delta\tau}{\partial x}\frac{\partial \tau_s}{\partial x} + \frac{\partial \delta\tau}{\partial z}\frac{\partial \tau_s}{\partial z} = \delta p(x,z) p_s(x,z), \tag{C-3}$$

where τ_s is the traveltime calculated by using the smoothed slowness; that is,

$$\left(\frac{\partial \tau_s}{\partial x}\right)^2 + \left(\frac{\partial \tau_s}{\partial z}\right)^2 = p_s^2(x,z). \tag{C-4}$$

Thus, the traveltime in the original slowness can be approximated by

$$\tau(x,z) \approx \tau_s(x,z) + \delta\tau(x,z). \tag{C-5}$$

Similarly, if traveltime is calculated by the ray tracing method, the traveltime perturbation can be represented by

$$\delta\tau = \int_L \delta p(x,z) dL, \tag{C-6}$$

where L is the raypath calculated by using the smoothed slowness. This perturbation technique works if the slowness perturbation is sufficiently small.

This idea also can be applied to other ray-based migration approaches, such as the Gaussian beam method.