Chapter 2

Beam-stack and interval velocity

2.1 OVERVIEW

Seismic velocity is estimated by searching for the interval-velocity function that best predicts the kinematics of the data decomposed with beam stack. Beam stack is a local stacking operator that provides a detailed information on the kinematics of the reflections by transforming the recorded data according to the ray parameter of the reflections. The first section of this chapter introduces the beam stack transform and illustrates the improvements in resolution achieved by transforming the data with beam stack instead of conventional local slant stack.

The traveltimes and the surface locations of the semblance peaks in the transformed data, are efficiently modeled using ray tracing. The velocity estimation can thus be formulated as an optimization process that maximizes the semblance of beam stacks at the traveltimes and surface locations modeled by ray tracing. The properties of the inversion problem depend on the characteristics of the forward modeling, that is, on the relations between the velocity model and the kinematics of the beam-stacked data. Therefore, at the end of the chapter, I examine the perturbations in beam stacks caused by a velocity anomaly.

2.2 DATA DECOMPOSITION USING BEAM STACK

For the purpose of determining interval velocity, stacking operators are often used for measuring the moveouts of the reflections. In particular, hyperbolic stacks are used for computing stacking velocity spectra (Taner and Koehler, 1969). Stacking velocity spectra are computed measuring the coherency of the data along hyperbolic trajectories in offset

and time. The hyperbolic trajectories are parametrized by the time of the apex of the hyperbola and by its curvature, that is, stacking velocity. When the traveltimes are not hyperbolic functions of offset, two parameters are not sufficient to describe the moveouts. In these cases the data can be decomposed with local stacking operators. The stacking trajectories of local stacking operators are parametrized by three parameters: the time $\bar{t}$ and half offset $\bar{h}$ of the central point of the trajectories, and by the "time dip" $p_h = dt/dh$ at the central point $(\bar{t}, \bar{h})$. The parameter p_h has the physical interpretation of being the ray parameter of the recorded reflections. Therefore a local stack decomposes a gather, at each traveltime and each offset, according to the ray parameter of the reflections. This property leads to an efficient algorithm for modeling the kinematics of data transformed with a local stack; I will present the modeling algorithm in section 2.3.1.

Local slant stack is the local stacking operator commonly used for analyzing seismic data (Hermont, 1979; Sword, 1987). Unfortunately, the resolution of local slant stack applied to CMP gathers is limited because slant stack's straight stacking trajectories do not well approximate the hyperbolic moveouts in CMP gathers. In this section, after reviewing local slant stack and illustrating its limitations, I introduce beam stack (Kostov and Biondi, 1987) as a family of local stacking operators that overcome the limitations of slant stack by stacking data along curved trajectories (hyperbolic or parabolic).

2.2.1 Local slant stack of common midpoint gathers

Local slant stack is computed by summing the data along straight segments. In a CMP gather the stacking trajectories of local slant stacks are the straight lines

$$\mathrm{T}_{p_h}(h) = \bar{t} + p_h(h - \bar{h}). \tag{2.1}$$

The slant stacked data at the point $(\bar{t}, \bar{h})$ are computed by use of the following expression

$$\mathrm{CMP}_{\mathrm{Stack}}(\bar{t}, \bar{h}, p_h) = \frac{1}{2L_h + 1} \sum_{j=-L_h}^{L_h} \mathrm{CMP}\left[\mathrm{T}_{p_h}(\bar{h} + j\Delta h), \bar{h} + j\Delta h\right], \tag{2.2}$$

where $(2L_h + 1)$ is the number of stacked traces, and Δh is the half-offset interval.

The expression in equation (2.2) is a linear transformation of the data. It has the advantage of being invertible (Trorey, 1961; Thorson, 1985; Kostov, 1989), but, for the purpose of velocity analysis, it has the disadvantage that the stacked results depend not

only on the coherency of the data along the stacking trajectories but also on the amplitude of the reflections. Semblance is often used to estimate the coherency of the data along the stacking trajectories independently from the data amplitudes (Neidell and Taner, 1971). Semblance is always positive and less than or equal to one; it is equal to one when the amplitudes of the data are all the same along the stacking trajectory, and it is zero when the data are completely uncoherent. When semblance is used as a coherency measure, the expression for computing the local slant stack of a gather is

$$\mathrm{CMP}_{\mathrm{Sem}}(\bar{t},\bar{h},p_h) = \frac{\left(\sum_{j=-L_h}^{L_h} \mathrm{CMP}\left[\mathrm{T}_{p_h}(\bar{h}+j\Delta h),\bar{h}+j\Delta h\right]\right)^2}{(2L_h+1)\sum_{j=-L_h}^{L_h} \mathrm{CMP}^2\left[\mathrm{T}_{p_h}(\bar{h}+j\Delta h),\bar{h}+j\Delta h\right]}. \qquad (2.3)$$

The resolution of local slant stack applied to a CMP gather is limited because the straight stacking trajectories do not well approximate the hyperbolic moveouts of the reflections. The "effective" length of local slant stack, that is, the length of the trajectories on which reflections are summed coherently, is limited to the Fresnel zone of the reflections (Claerbout, 1975). This limitation affects the resolution that can be achieved using local slant stack because the resolution of stacking operators is proportional to their "effective" length (Appendix A). Figure 2.1 shows the effects of moveouts' curvature on the results of local slant stack. The figure shows a CMP gather from the Adriatic Sea data set and the semblance functions obtained by local slant stacking the gather. I slant stacked the gather using two different lengths for the stacking trajectories: 550 m for computing the panel in the center and 1050 m for the panel on the right. Increasing the stacking trajectory length has not improved the result, as it would have happened if the actual "effective" length had been longer, but, on the contrary, it has increased the smear of the semblance peaks. To improve the resolution of local coherency measures I replace local slant stack with beam stack.

2.2.2 Beam stack of common midpoint gathers

The resolution of local stacking operators can be improved by use of curved stacking trajectories, which approximate the moveouts of the reflections better than straight trajectories. Beam stacks are local stacking operators with curved stacking trajectories (hyperbolic or

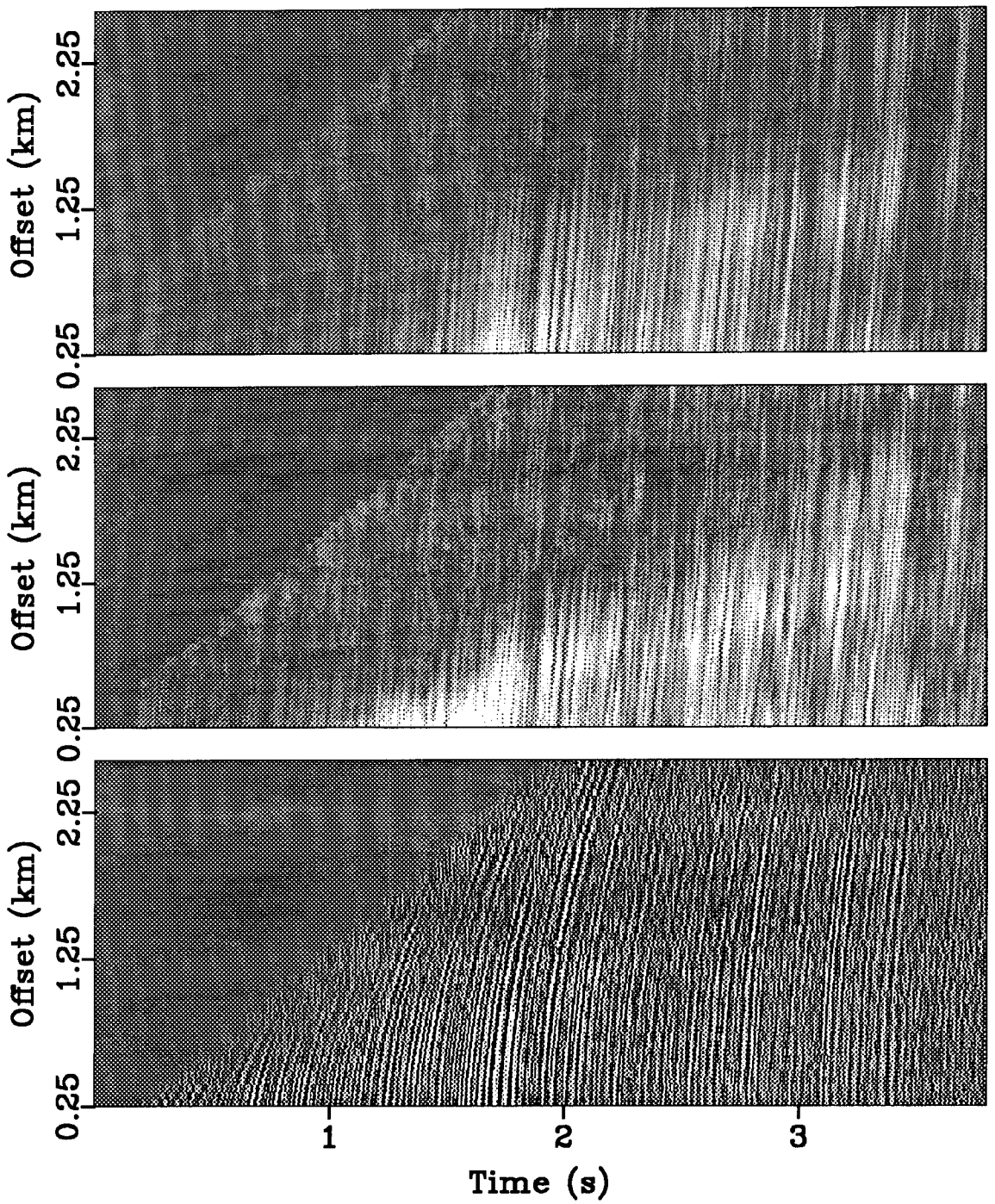

FIG. 2.1. A CMP gather (left) from the Adriatic Sea data set and the semblance functions obtained by slant stacking the gather. The stacking trajectories were 550 m long (center) and 1050 m long (right). The offset ray parameter p_h was equal to 0.1 s/km.

parabolic). In general the curvature of the traveltime curves depends on the velocity of propagation of the reflections and on the reflectors' geometry. This dependence can be a problem if we want to use the result of beam stacks for estimating velocity. In this section I show that the curvature can be approximately estimated with little information on velocity, given the traveltime, the offset, and the ray parameter.

Hyperbolic beam stack

In many interesting cases the reflections in a CMP gather can be well approximated by Dix hyperbolas independently from the velocity and the reflectors' geometry. In these cases the traveltime curves belong to the family of hyperbolas defined by the equation

$$t^2 = t_0^2 + \frac{4h^2}{V^2}, \tag{2.4}$$

where the time of the apex t_0 and the velocity V are free parameters. A hyperbola of this family is completely determined when the slope p_h of the tangent at the traveltime $\bar{t}$ and the half offset $\bar{h}$ is fixed; that is, the stacking trajectory is completely determined by $\bar{t}$, $\bar{h}$, and p_h.

The derivative of traveltime t with respect to the half offset h is equal to the ray parameter p_h; that is

$$\left.\frac{dt}{dh}\right|_{h=\bar{h}} = p_h = \frac{4\bar{h}}{V^2\bar{t}}. \tag{2.5}$$

This relation and equation (2.4) can be used for determining the velocity squared,

$$V^2 = \frac{4\bar{h}}{p_h\bar{t}}, \tag{2.6}$$

and the time of the apex squared,

$$t_0^2 = \bar{t}^2 - \bar{h}\bar{t}p_h, \tag{2.7}$$

as functions of $\bar{t}$, $\bar{h}$, and p_h. Substituting these values in the equation of the Dix hyperbolas

[equation (2.4)], we find the equation for the hyperbolic stacking trajectories of beam stack

$$\mathrm{T}_{p_h}(h) = \sqrt{\bar{t}^2 - \bar{h}\bar{t}p_h + \frac{h^2}{\bar{h}}p_h\bar{t}}. \tag{2.8}$$

Hyperbolic beam stack is computed substituting the stacking trajectory of equation (2.8) in the stacking equation (2.2), or in the semblance equation (2.3).

Figure 2.2 shows the improvements in resolution when beam stack instead of slant stack is used. The upper part of Figure 2.2 shows the semblance functions computed by the application of slant stack to the CMP gather displayed in Figure 2.1. Three different lengths of the stacking trajectory were used. The resolution of the slant stacks does not improve, it actually degrades, with the increasing length of the stacking trajectory, because of the effect of the curvature of the reflections, as predicted by equation (A.12). On the contrary the resolution of beam stacks, which are shown in the lower part of Figure 2.2, is not limited by the curvature of the reflections but improves as the stacking trajectories become longer.

Hyperbolic beam stack is a time-variant transformation and thus cannot be implemented in the frequency domain. In the time domain the most efficient way of implementing beam stack is precomputing the operator and then applying it to many CMP gathers at the same time.

Parabolic beam stack

Hyperbolas can be locally approximated by parabolas. Therefore the following parabolic approximation of the hyperbolic trajectories can be used for computing beam stack

$$\mathrm{T}_{p_h}(h) = \bar{t} + p_h(h - \bar{h}) + \frac{\rho(h - \bar{h})^2}{2}, \tag{2.9}$$

where ρ is the second derivative of traveltime with respect to h and it is related to the curvature of the reflections.

When the Dix approximation is valid ρ can be easily derived from equation (2.5)

$$\rho = \frac{d^2t}{dh^2} = \frac{4}{V^2\bar{t}} = \frac{p_h}{\bar{h}}; \tag{2.10}$$

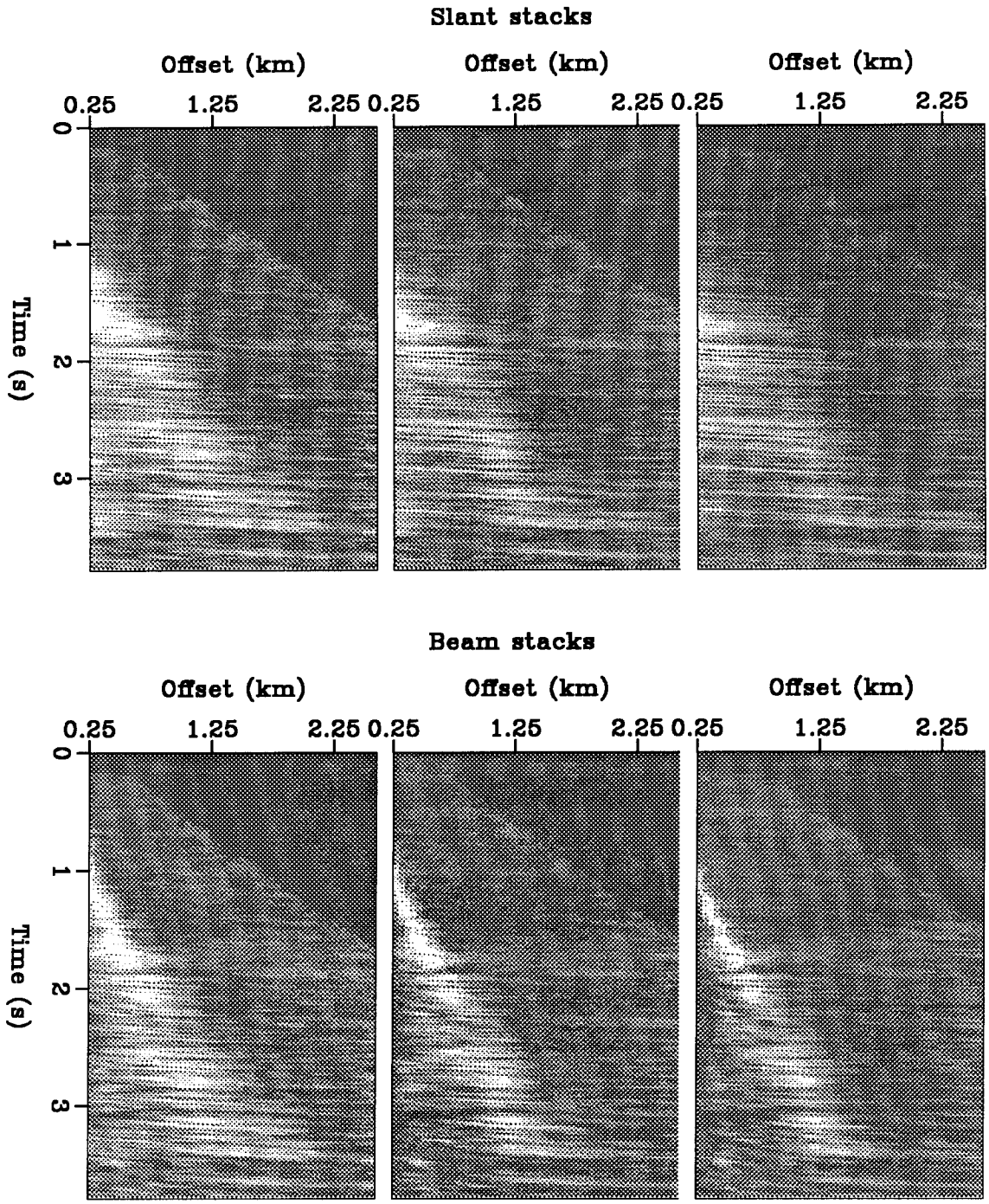

FIG. 2.2. Comparison of beam stacks (bottom) with local slant stacks (top). The semblance panels were computed using three different lengths of the stacking trajectory: 550 m (left), 850 m (center), and 1050 m (right). The resolution of beam stacks improves as the length of the stacking trajectory is increased, while the results of slant stacks degrade.

it can be substituted in equation (2.9) to give the stacking trajectory

$$T_{p_h}(h) = \bar{t} + p_h(h - \bar{h}) + \frac{p_h(h - \bar{h})^2}{2\bar{h}}. \tag{2.11}$$

The parabolic beam stack defined in equation (2.11) is a time-invariant transformation and therefore it can more easily be efficiently implemented than hyperbolic beam stack. Considering the fact that the differences in the results obtained with parabolic instead of hyperbolic beam stack on real data are insignificant, I often used parabolic beam stack in processing real data.

Another theoretical advantage of parabolic beam stack is that it can be used when the traveltime curves are not well approximated by Dix hyperbolas. If an a priori velocity model is available, the second derivative of traveltime with respect to the spatial axis can be estimated by dynamic ray tracing (Červený, 1987) and the data can be beam-stacked using equation (2.9). The beam-stacked data can then be used for improving the velocity model.

2.2.3 Velocity resolution and beam stack parameters

When beam stack is used to decompose a data set there are many parameters to be set. In this section I discuss the criteria for choosing two of the most critical parameters: the sampling interval of offset ray parameter Δp_h and the length of the beam stack's trajectories L_h. The choices of these two parameters are not independent and are related to the quality of the data (signal-to-noise ratio, frequency, maximum offset) as well as to the desired spatial resolution of the velocity estimation.

When there are no lateral variations in velocity, and thus the moveouts are hyperbolic, the velocity can be reconstructed from data decomposed using only one offset ray parameter. When there are velocity variations, the more offset ray parameters that are used the more precisely the moveouts in the data are measured, and consequently the better the velocity variations can be estimated. However, in practice, beam stacks with different offset ray parameters are correlated in a measure proportional to the overlap of the corresponding stacking trajectories. If the stacking trajectories are long, beam stacks computed for slightly different ray parameters are strongly correlated. Therefore the shorter the stacking trajectories, the smaller the difference Δp_h between independent beam stacks. On the other hand the resolution of the beam stack transformation decreases

with the length of the stacking trajectories [equation (A.6)]. A lower limit for the length of the stacking trajectories is reached when the uncertainties on the location of the peak of beam stack is larger than the length of stacking trajectories. Furthermore the longer the stacking trajectories the less sensitive is the beam stack to noise in the data. The correct balance between resolution and reliability of the estimation is data dependent and must also take in account the cost factor. The cpu time and the data storage required by the velocity-estimation procedure are proportional to the number of ray parameters.

From the previous considerations it is clear the advantage of increasing the resolution of beam stack without increasing the stacking trajectory length. In Chapter 4 I present a non-linear coherency criterion that could be used, instead of the classical stacking estimator, for increasing the resolution of local measures of moveout parameters.

The resolution of beam stack is not only dependent on the length of the stacking trajectory but it is also proportional to the frequency of the data and to the curvature of the wavefronts [equation (A.6)]. Therefore it can be useful to change the length of the stacking trajectory with time to compensate for the variations of frequency and curvature, and achieve the same resolution for the whole gather. In particular it can be useful to increase the length of the stacking trajectories with time, because the curvature of the wavefronts decreases with time [equation (2.10)] and the frequency of the reflections decreases with traveltime because of frequency-dependent attenuation.

2.2.4 Two-dimensional beam stack

When there are dipping reflectors the moveouts of the reflections along the offset axis are not sufficient to determine interval velocity. Information on the kinematics of the reflections along the midpoint direction is also needed and it can be extracted from the data by applying local slant stacks to common-offset sections. The expression for decomposing a common-offset section $\mathrm{Coff}(t,y)$ along the midpoint direction y by use of semblance as coherency criterion is

$$\mathrm{Coff}_{\mathrm{Sem}}(\bar{t},\bar{y},p_y) = \frac{\left(\sum_{k=-L_y}^{L_y} \mathrm{Coff}\left[\mathrm{T}_{p_y}(\bar{y}+k\Delta y),\bar{y}+k\Delta y\right]\right)^2}{(2L_y+1)\sum_{k=-L_y}^{L_y} \mathrm{Coff}^2\left[\mathrm{T}_{p_y}(\bar{y}+k\Delta y),\bar{y}+k\Delta y\right]}, \tag{2.12}$$

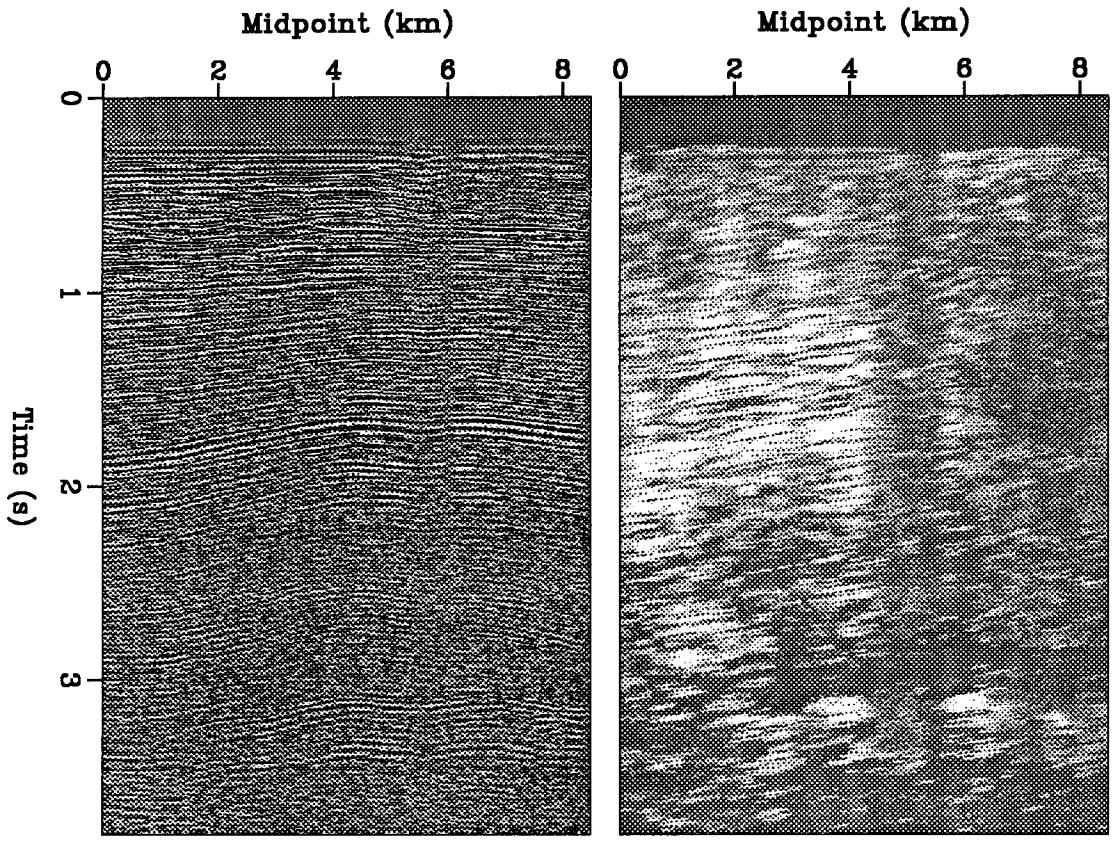

FIG. 2.3. A constant-offset section of the data set from the Adriatic Sea and the semblance panel computed by slant stacking the section with p_y equal to 0.04 s/km.

where $(2L_y + 1)$ is the number of stacked traces, Δy is the midpoint interval, and the stacking trajectories are the straight lines

$$\mathrm{T}_{p_y}(y) = \bar{t} + p_y(y - \bar{y}). \tag{2.13}$$

Figure 2.3 shows a constant-offset section from the Adriatic Sea data set and the semblance panel computed by slant stacking the same section with p_y equal to 0.04 s/km.

Prestack data can be simultaneously decomposed according to both the midpoint ray parameter and the offset ray parameter; this can be done with a two-dimensional local stack (Harlan and Burridge, 1983; Pan and Gardner, 1986). Two-dimensional stacking operators are similar to the one-dimensional operators of equations (2.3) and (2.12) but the data coherency is measured on surfaces instead of over lines. The mathematical expression for

a two-dimensional beam stack is

$$\mathrm{Beam}(\bar{t},\bar{y},\bar{h},p_y,p_h) = \frac{1}{(2L_y+1)(2L_h+1)}\times$$
$$\frac{\left(\sum_{k=-L_y}^{L_y}\sum_{j=-L_h}^{L_h}\mathrm{Data}\left[\mathrm{T}_{p_y}(\bar{y}+k\Delta y)+\mathrm{T}_{p_h}(\bar{h}+j\Delta h)-\bar{t},\bar{y}+k\Delta y,\bar{h}+j\Delta h\right]\right)^2}{\sum_{k=-L_y}^{L_y}\sum_{j=-L_h}^{L_h}\mathrm{Data}^2\left[\mathrm{T}_{p_y}(\bar{y}+k\Delta y)+\mathrm{T}_{p_h}(\bar{h}+j\Delta h)-\bar{t},\bar{y}+k\Delta y,\bar{h}+j\Delta h\right]}, \quad (2.14)$$

where the trajectories T_{p_y} are straight [equation (2.13)], and the trajectories T_{p_h} are curved [equation (2.8) or equation (2.11)].

The two-dimensional beam stack is more expensive to compute than the one-dimensional one and, moreover, it requires the storage of the results as a five-dimensional data set. Because of these practical reasons I approximate the two-dimensional beam stack as the product of two, one-dimensional stacks. That is:

$$\mathrm{Beam}(\bar{t},\bar{y},\bar{h},p_y,p_h) \approx \mathrm{Coff}_{\mathrm{Sem}}(\bar{t},\bar{y},\bar{h},p_y)\times\mathrm{CMP}_{\mathrm{Sem}}(\bar{t},\bar{y},\bar{h},p_h). \quad (2.15)$$

The main advantage of the approximation is that the product of the results of the one-dimensional stacks can be computed on the fly during the velocity estimation process; thus, two four-dimensional data sets can be stored on disk instead of a five-dimensional data set. On the other hand the results of a two-dimensional beam stack have a higher signal-to-noise ratio because their computation uses more data samples.

2.3 PRINCIPLES OF VELOCITY ESTIMATION

The basic principle of velocity estimation is to find the velocity function that used for modeling best predicts the data transformed by beam stack. In particular, the estimated velocity must explain the kinematics of beam-stacked data. Therefore the first part of this section presents how the kinematics of the beam-stacked data can be efficiently modeled by ray tracing. Simplicity and lower computational cost comprise the rationale for using ray tracing. A practical problem of using ray tracing for modeling the data is that the solution of the ray equations is quite unstable when the velocity function is not smooth. The solution to this problem is allowing only smooth velocity functions; this is accomplished

by a proper parametrization of velocity, as I will discuss in section 2.3.3.

The results of modeling can be matched to the kinematics of the transformed data according to different criteria. A simple solution would be to pick the traveltimes and surface locations of the semblance peaks in the beam-stacked data, and then to minimize the differences between the picked values and the values computed by ray tracing. Although attractive, this solution has the danger of relying on the quality of the preliminary picking of the data. Coherent noise, as for example multiples, can cause spurious peaks in the transformed data. If these spurious peaks are picked, they will influence the final solution. Therefore, instead of picking the data I maximize the energy in the beam-stacked data at the traveltimes and surface locations predicted by ray tracing. This approach corresponds to a model-driven detection of the events to be matched by the estimation. The semblance peaks that cannot be fitted by applying small perturbations to the current velocity model do not influence the estimation process.

In the second part of the section the criterion used for evaluating the quality of a velocity model is formalized as an objective function of an optimization problem. The algorithm for maximizing this objective function is presented in Chapter 3.

2.3.1 Using ray tracing to model beam-stacked data

A beam stack can be efficiently modeled with an initial-value ray tracing method because the initial conditions at the surface are specified by the values of the ray parameters used for the data decomposition. Initial-value ray tracing is more efficient than two-points ray tracing because it requires the numerical solution of a system of ordinary differential equations fixing the initial conditions instead of the boundary conditions. Furthermore, ray tracing from the surface down into the earth (top-down) is much cheaper than ray tracing from the reflector up to the surface (bottom-up). Because the surface locations lie on a line, a top-down ray tracing needs to trace fewer rays than a bottom-up ray tracing, for which the starting points fill a two-dimensional space.

Beam stack decomposes the data according to the offset ray parameter p_h and the midpoint ray parameter p_y, while ray tracing needs the ray parameters in field coordinates, that is, the shot ray parameter p_s and the receiver ray parameter p_r. The expressions that relate p_s and p_r used for ray tracing to p_y and p_h used for decomposing the data are derived in Appendix B.

Given the velocity model $\mathbf{m}$, I use the ray tracing algorithm presented in Appendix C

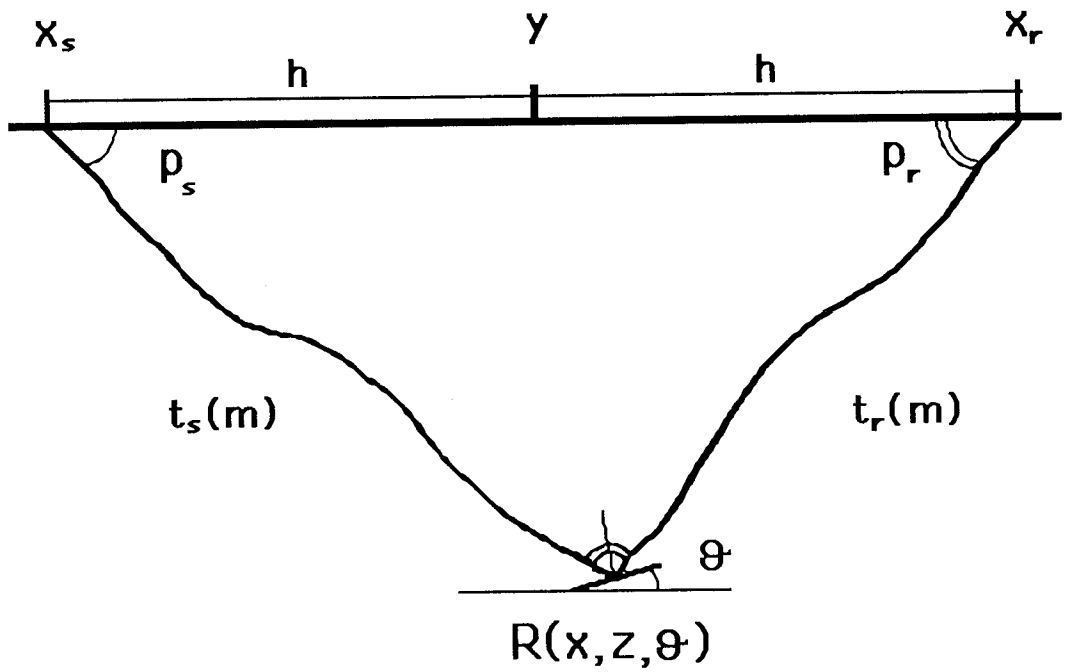

FIG. 2.4. The paths of the down-going ray and of the up-going ray meeting at the reflector $R(x, z, \theta)$.

for tracing the rays for each ray parameter p_s and p_r and from each surface location. For computing the traveltime and surface locations of the beam stacks, I then combine the rays following the geometry shown in Figure 2.4. The shot-ray is traced downward from the source location x_s starting with the horizontal ray parameter p_s, and the receiver-ray is traced from the receiver location x_r starting with horizontal ray parameters p_r. The traveltime along the shot-ray is t_s and the traveltime along the receiver-ray is t_r. The meeting point of the shot-ray and of the receiver-ray is the estimated reflector $R(x, z, \theta)$, where x and z are the spatial coordinates of the reflector and θ is its dip.

The total traveltime t, midpoint y, and half-offset h are, respectively,

$$t(\mathbf{m}) = t_s(\mathbf{m}) + t_r(\mathbf{m}); \tag{2.16}$$

$$y = \frac{1}{2}(x_s + x_r); \tag{2.17}$$

and

$$h = \frac{1}{2}(x_r - x_s). \tag{2.18}$$

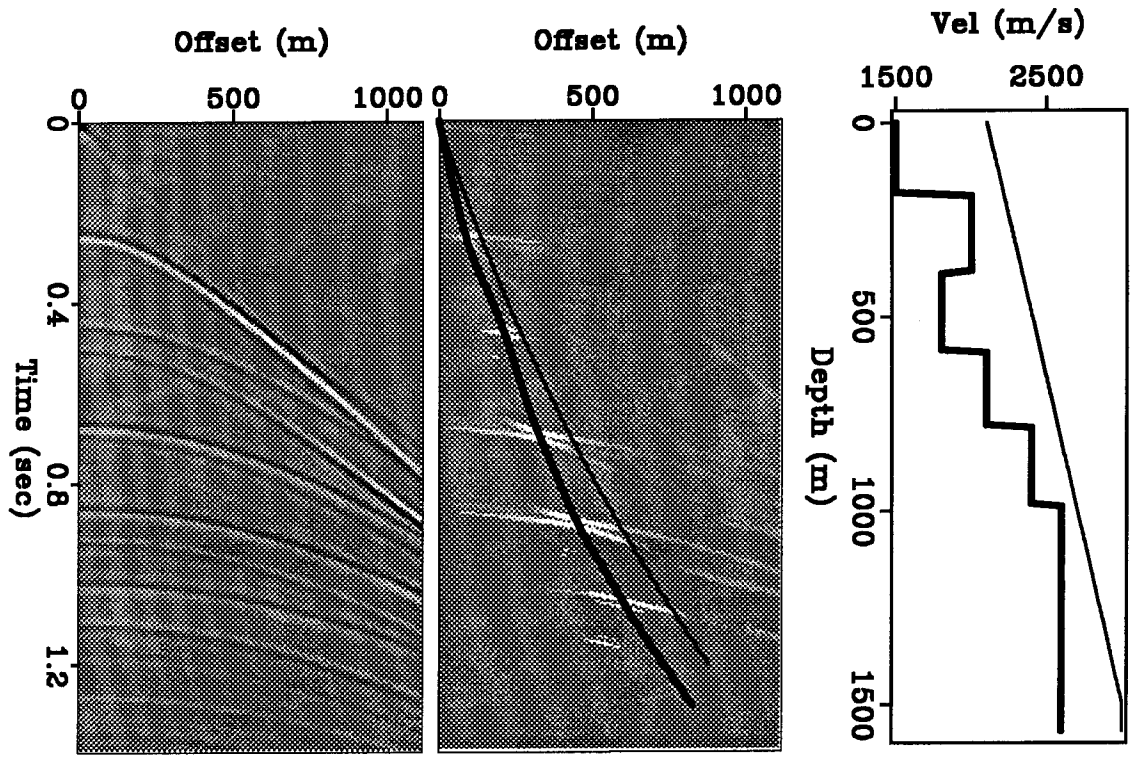

FIG. 2.5. A synthetic CMP gather and a slice of the semblance cube taken at constant offset ray parameter p_h=.6 s/km. The traveltime curves as a function of offset, superimposed to the beam-stacked data, are computed assuming two different velocity functions. The true velocity was assumed for computing the thicker line while a too high velocity was assumed for computing the thinner line.

These relations define the traveltime function

$$t = t(y, h, p_y, p_h, \mathbf{m}), \tag{2.19}$$

that represents a manifold in the data space. The manifold is a function of the velocity model $\mathbf{m}$ and it represents the results of modeling with ray tracing. Each point $(\bar{t}, \bar{y}, \bar{h}, \bar{p}_y, \bar{p}_h)$ on the manifold corresponds to a reflector $R(\bar{x}, \bar{z}, \bar{\theta})$ in the image space, and thus the results of ray tracing can be used for imaging the reflectors from the beam-stacked data Beam(t, y, h, p_y, p_h). The contribution of the data point $(\bar{t}, \bar{y}, \bar{h}, \bar{p}_y, \bar{p}_h)$ to the image at $(\bar{x}, \bar{z}, \bar{\theta})$ is equal to Beam$(\bar{t}, \bar{y}, \bar{h}, \bar{p}_y, \bar{p}_h)$. The whole image is computed summing the contributions from all the data points on the manifold.

Figure 2.5 shows a simple example of a beam stack's modeling by ray tracing under the assumptions that the earth is horizontally layered. The left side of Figure 2.5 shows

a CMP gather modeled by finite difference, with the assumed velocity function shown by the thicker line on the right side of the figure. When the earth is horizontally layered the reflections are independent of the midpoint coordinate and thus their midpoint ray parameter p_y is null. The semblance data computed by use of beam-stacks reduce to a three-dimensional data cube Beam$(t, y = y_0, h, p_y = 0, p_h)$, and the manifold of equation (2.19) becomes a surface in a three-dimensional space. The right side of Figure 2.5 shows a slice of the semblance cube taken at constant offset ray parameter p_h=.6 s/km. The traveltime curves as a function of offset, superimposed to the beam-stacked data, are computed assuming two different velocity functions. The thicker line corresponds to the velocity function used for modeling the data, while the thinner line corresponds to the higher velocity function drawn with a thinner line in Figure 2.5. When the assumed velocity is correct, the ray modeling predicts correctly the position of the semblance peaks in beam-stacked data; when the assumed velocity is too high, the predicted offsets are too large.

This simple example shows that the sum of the beam stacks' energy along the traveltime curves is a good criterion for the evaluation of a velocity model. If the velocity along the raypath is correct, the traveltime curves pass close to the semblance peaks and the total sum increases; otherwise the traveltime curves pass away from the peaks and the sum is not augmented.

2.3.2 Maximization of beam stacks' energy

The criterion for judging the correctness of a velocity function suggested by the previous example can be generalized to handle the more complex cases of dipping reflectors and lateral velocity variations. If Beam(t, y, h, p_y, p_h) are the transformed data, and $t(y, t, p_y, p_h, \mathbf{m})$ is the result of ray tracing in a general 2-D velocity model $\mathbf{m}$, the estimation problem can be formulated as the non-quadratic optimization problem of finding the maximum with respect to $\mathbf{m}$ of the following sums:

$$E(\mathbf{m}) = \sum_{y} \sum_{h} \sum_{p_y} \sum_{p_h} \text{Beam}\left(t(y, h, p_y, p_h, \mathbf{m}), y, h, p_y, p_h\right). \tag{2.20}$$

When an a priori estimate $\mathbf{m_0}$ of the velocity model and of its covariance matrix $\mathbf{C_m}$

exists, the objective function becomes

$$Q_t(\mathbf{m}) = \sum_y \sum_h \sum_{p_y} \sum_{p_h} \text{Beam}\left(t(y,h,p_y,p_h,\mathbf{m}),y,h,p_y,p_h\right) - (\mathbf{m}-\mathbf{m_0})^T \mathbf{C_m}^{-1}(\mathbf{m}-\mathbf{m_0}). \tag{2.21}$$

The objective function defined in equation (2.21) is highly non-quadratic. Ray tracing is not a linear function of the velocity model, and thus the modeling function $t(y,h,p_y,p_h,\mathbf{m})$ is not linear. Furthermore the fitting function, represented by the beam stacks' semblance $\text{Beam}(t,y,h,p_h,p_y)$, is not a parabolic function of the traveltime t. Semblance can be well approximated by a parabolic function only in the proximity of its peaks. In Chapter 3 I will present gradient algorithms for solving this difficult maximization problem.

2.3.3 Smooth parametrization of the velocity model

The velocity estimation is based on the kinematics of the reflections and thus it can resolve only the low-wavenumbers of the velocity function. Traveltimes are integral measures of velocity along the raypaths and consequently full ray coverage of the velocity model is necessary for resolving all spatial frequencies in the model. The ray coverage is usually poor in a reflection seismic experiment because the sources and receivers are constrained at the surface and the reflectors have a finite range of dips. If no a priori information is available on the high spatial frequency of the velocity model the safest assumption is to constrain them to be zero. Another reason for constraining the high spatial frequency of the velocity model is that the waves propagating in the earth are band limited in frequency; while ray theory is a high frequency approximation of wave propagation. The propagation of band-limited waves is not influenced only by the velocities along the raypaths; it is also affected by velocities averaged in a region called the Fresnel zone (Woodward, 1989). This spatial averaging caused by the frequency limitation of the reflections is not taken into account by a tomographic back projection along raypaths. The velocity estimation is therefore unstable because it is based on a wrong assumption on the spatial resolution of the data.

A common approach for constraining poorly determined components of the velocity model is to impose a smoothness condition using a derivative operator as the inverse of the model covariance matrix $\mathbf{C_m}$ (Toldi, 1985). This approach is straightforward and easily

implemented but it has the disadvantage that the optimization problem is solved for many more unknown parameters than are needed to describe a smooth velocity model. The cost of the estimation procedure is therefore higher than necessary.

I use a parametrization of the velocity function that takes the smoothness condition into account implicitly. I parametrize the velocity function with smooth basis functions; in particular I use B-spline basis functions that have a continuous second derivative. This parsimonious parametrization of the model allows to formalize the estimation problem in function of only the strictly necessary number of unknowns. This reduction makes possible to store on disk the linear operator that relates model parameters to the data and consequently saves a considerable amount of computer time (section 3.3). Appendix D contains the details on the model parametrization by B-splines.

2.4 EFFECTS OF A VELOCITY ANOMALY ON BEAM-STACKED DATA

Conventional stacking velocity analysis could be used for estimating interval velocity when the velocity function changes slowly compared with the length of the cable, and the reflectors are flat. The tomographic methods, as the one presented in this thesis, become necessary when velocity changes considerably within the span of a cable and the reflectors are dipping. Therefore it is interesting to study the effects of a velocity anomaly, which is narrower than a cable length, on the beam stacks of the reflections from a dipping reflector. These effects are the information that the estimation method uses to reconstruct the velocity function. For this study I modeled a synthetic data set by a finite-difference program for propagating acoustic waves. I will first examine the beam-stacked data and explain the effects of the anomaly on the traveltimes and offsets of the beam-stacks' peaks. Then, because the estimation method uses ray tracing to model the behavior of beam stacks, I will compare the perturbations caused by the velocity anomaly on the finite-difference data with the perturbations that are predicted by ray tracing. Finally I will examine the traveltimes and offset perturbations caused by the anomaly and discuss the implications for the velocity estimation.

To generate the synthetic data set I modeled an off-end seismic survey with the geophones on the right side of the shots; the assumed velocity model is shown in Figure 2.6. The background velocity is 2.5 km/s and the circular velocity anomaly is a Gaussian

function with a peak velocity of 2.8 km/s. The velocity anomaly is estimated with the reflections from a bed dipping at 20°. I sorted the data in 175 common midpoint gathers (CMP), spaced 20 m. The minimum offset is 20 m and the maximum offset is 1500 m with offset sampling of 40 m. Figure 2.7 shows the nearest-offset section of the data set. The time pull-out in the middle of the section is caused by the velocity anomaly.

Figure 2.8 shows the gather recorded at the midpoint location of 1700 m and the semblance panels computed by beam stacking the gather according to two different offset ray parameters, $p_{h,1}$ equal to .124 s/km and $p_{h,2}$ equal to .158 s/km. The curves superimposed on the semblance plots show the traveltime as a function of offsets, modeled by ray tracing, when the velocity is constant and equal to the true background velocity. The traveltime curves pass through the semblance peaks because the reflections recorded at this specific midpoint location are not affected by the anomaly.

Figure 2.9 shows the gather recorded at the midpoint location of 2950 m and the corresponding semblance panels. The reflections recorded at this midpoint location have been perturbed by the anomaly and therefore the traveltime curves computed under the assumption of a constant velocity model (thinner lines) do not pass through the semblance peaks. On the contrary the curves computed assuming the true velocity (thicker lines) pass through these peaks. Figure 2.10 shows a graphic explanation of the perturbed offsets being smaller than the unperturbed ones for this particular CMP gather. The unperturbed raypaths (solid lines) and the perturbed raypaths (dashed lines) are constrained to have the same offset ray parameters and the same midpoint. The raypath of the up-going ray passes above the center of the anomaly and deviates away from it because the velocity perturbation is positive. The combination of the raypath perturbation and of the geometric constraints causes the offset of the perturbed reflection to be larger than the offset of the unperturbed reflection. The anomaly also causes a perturbation in the midpoint ray parameter of the reflections; that is, the time dips in the midpoint direction. This effect is related to the time pull-up caused by the anomaly.

Figure 2.11 shows another gather close to the anomaly, recorded at the midpoint location of 3450 m, and the corresponding semblance panels. For this midpoint location the perturbed offset of beam-stacks is smaller than the unperturbed one because the up-going raypath passes below the center of the anomaly instead of above it (Figure 2.12). The raypath of the up-going ray still deviates away from the positive velocity anomaly, but in

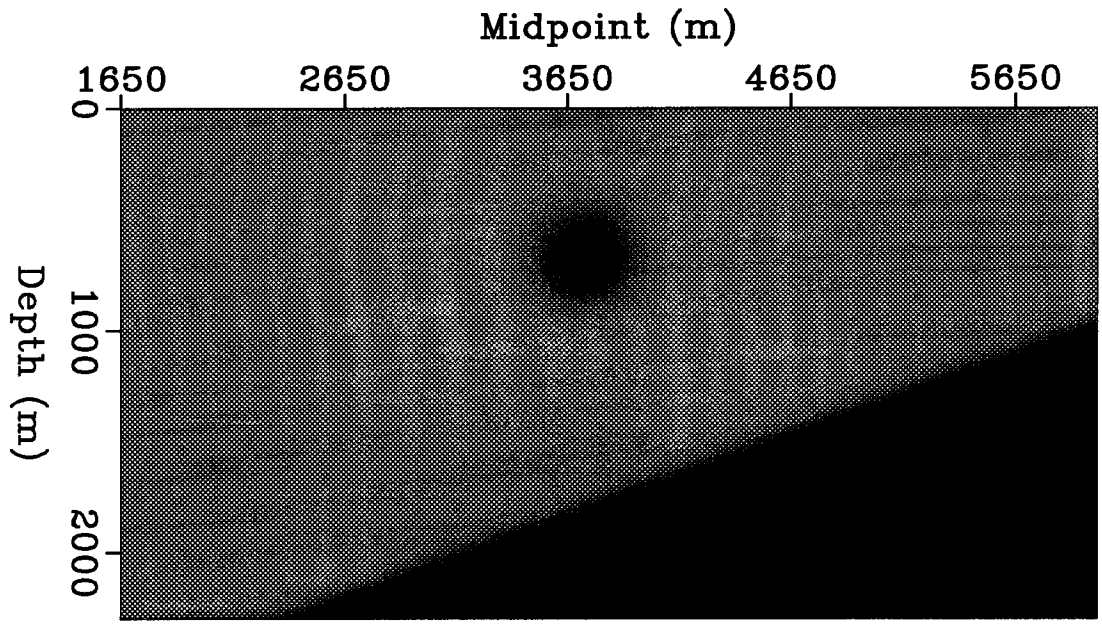

FIG. 2.6. The velocity model used to model the synthetic data. The background velocity is 2.5 km/s and the circular velocity anomaly is a Gaussian function with peak velocity of 2.8 km/s. The dipping reflector has a dip angle of 20°.

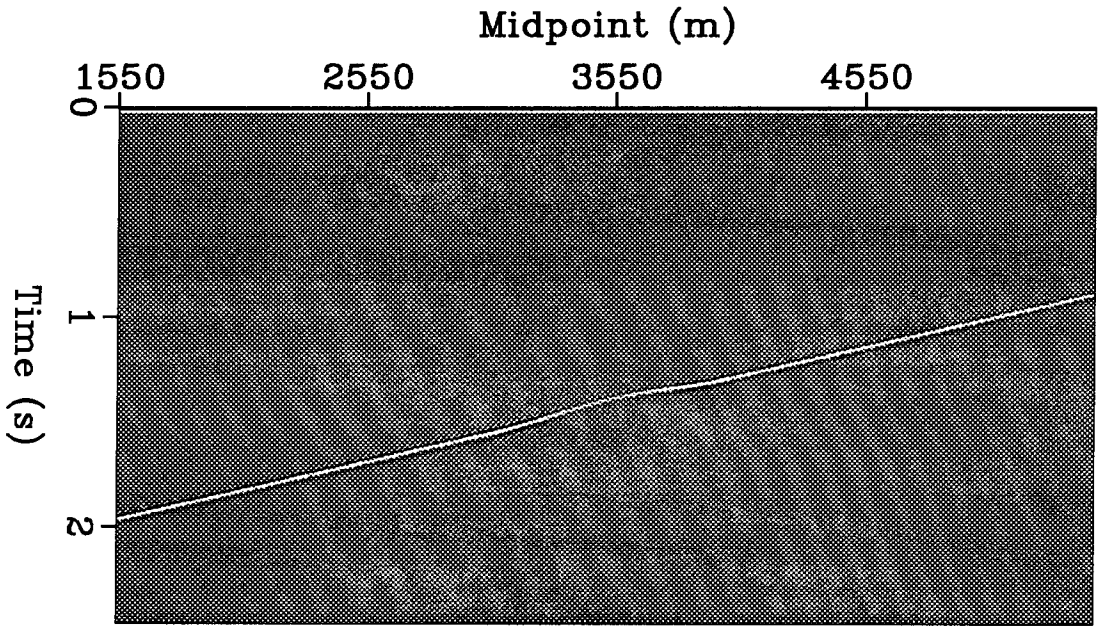

FIG. 2.7. The nearest-offset section of the synthetic data set. The time pull-out in the middle of the section is caused by the velocity anomaly.

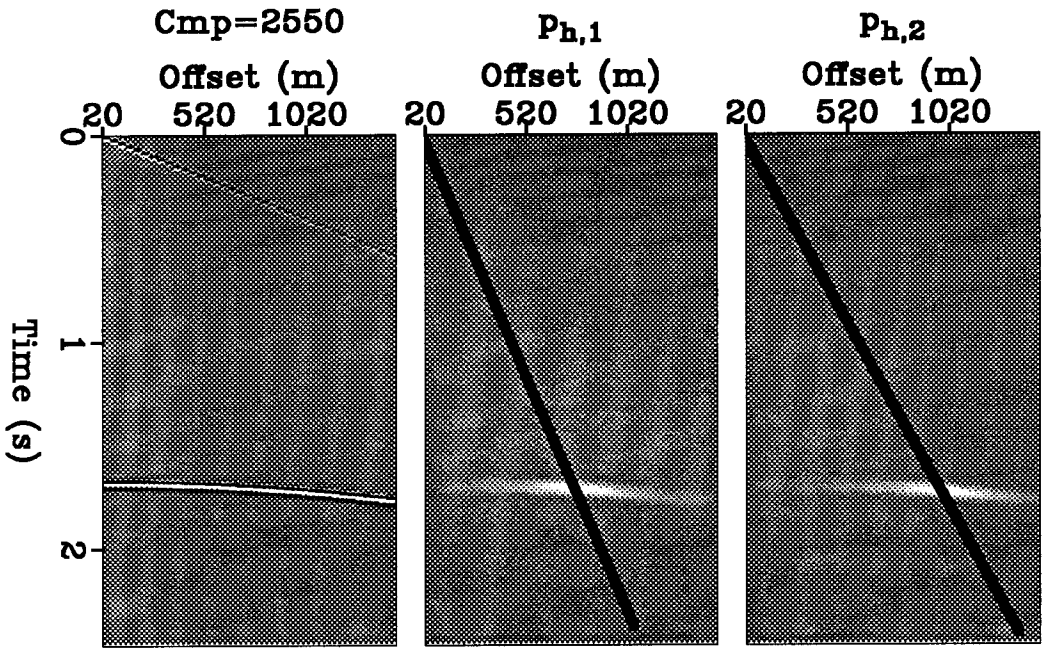

FIG. 2.8. The CMP gather recorded at the midpoint location of 1700 m and the semblance panels computed by beam stacking the gather according to two different offset ray parameters. The curves superimposed on the semblance plots show the traveltime as a function of offsets computed under the assumption of a constant velocity of 2.5 km/s. The traveltime curves pass through the beam-stacks' peaks because the reflections recorded at this midpoint location have not been perturbed by the velocity anomaly.

this case the perturbation causes a decrease in the offset instead of an increase. The perturbation in midpoint ray parameter is also opposite in sign compared with the previous case.

Figure 2.13 shows the offsets of beam-stacks peaks as a function of midpoint location for all the midpoints in the data set. The lower line shows the offset corresponding to the offset ray parameter $p_{h,1}$, while the upper line shows the offset corresponding to the offset ray parameter $p_{h,2}$. The positive perturbations in offset around the midpoint 2900 are caused by the up-going rays passing above the center of the anomaly. The negative perturbations around the midpoint 3550 are caused by both the up-going rays and the down-going rays passing below the center of the anomaly, while the second positive lobe is caused by the down-going rays passing above the center of the anomaly. The traveltimes of beam-stacks' peaks as a function of midpoint location are shown in Figure 2.14. The traveltime perturbations have a pattern similar to the offset perturbations but their relative amplitudes are smaller.

The velocity estimation method uses ray tracing to model the behavior of beam-stacked

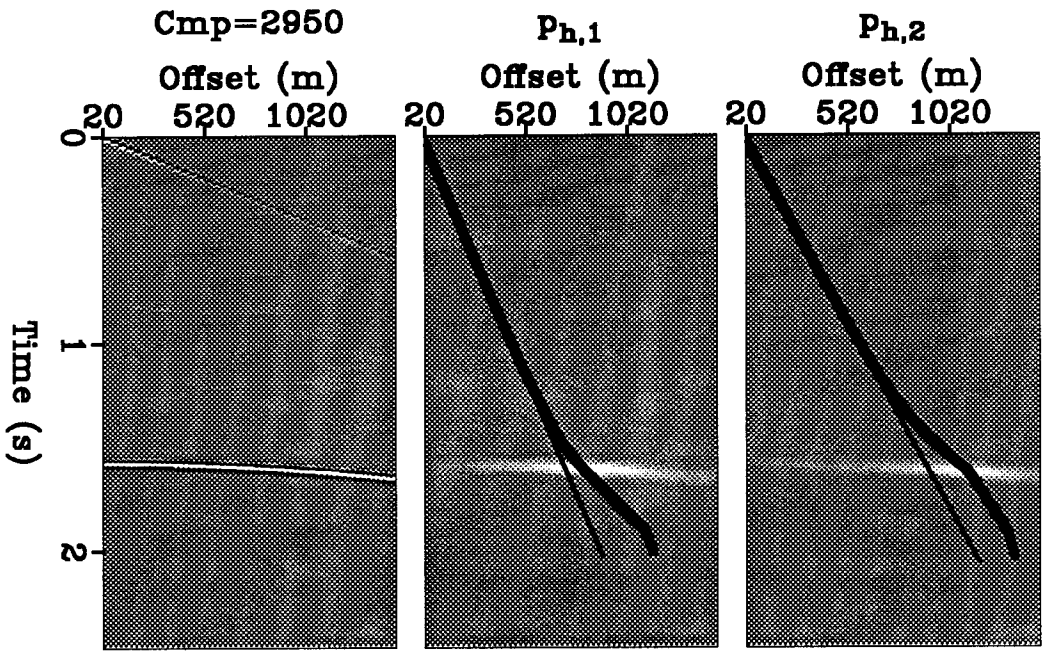

FIG. 2.9. The CMP gather recorded at the midpoint location of 2950 m and the semblance panels computed by beam stacking the gather according to two different offset ray parameters. The curves superimposed on the semblance plots show the traveltime as a function of offsets computed under the assumption of a constant velocity of 2.5 km/s (thinner lines) and under the assumption of the true velocity (thicker lines).

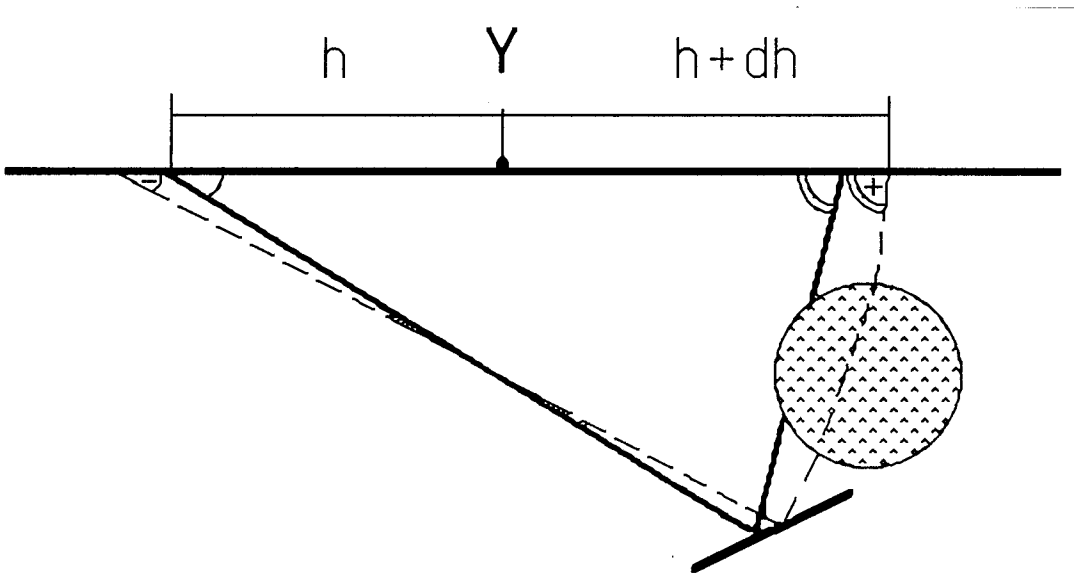

FIG. 2.10. The offset of the perturbed reflection (dashed line) is larger than the offset of the unperturbed reflection (solid line) because the up-going ray passes above the center of the anomaly and deviates away from it.

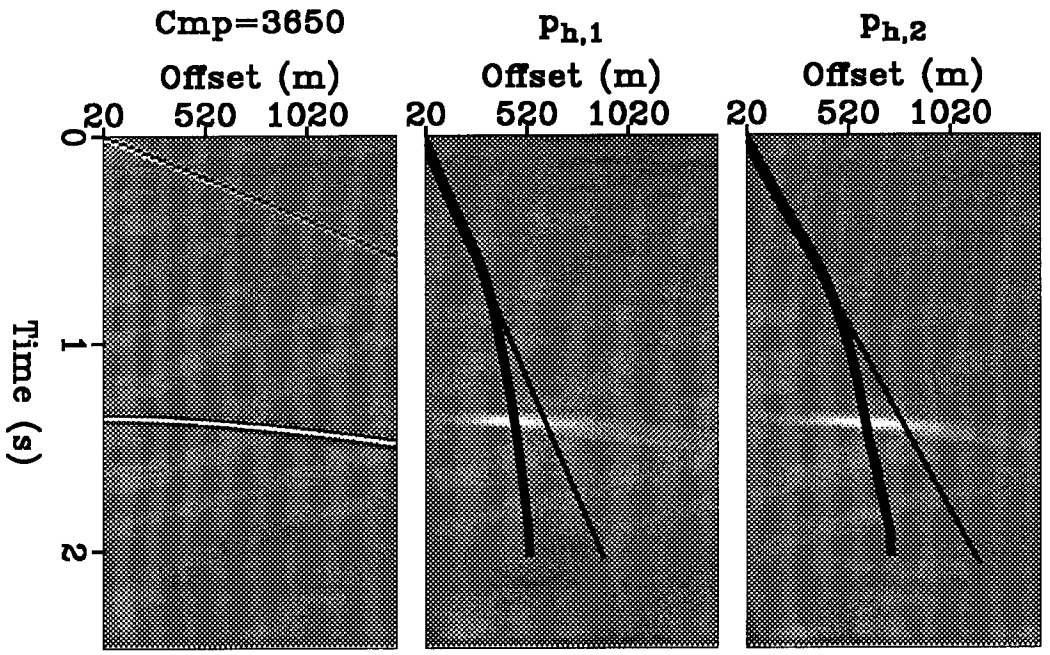

FIG. 2.11. The CMP gather recorded at the midpoint location of 3450 m and the semblance panels computed by beam stacking the gather according to two different offset ray parameters. The curves superimposed on the semblance plots show the traveltime as a function of offsets computed under the assumption of a constant velocity of 2.5 km/s (thinner lines) and under the assumption of the true velocity (thicker lines).

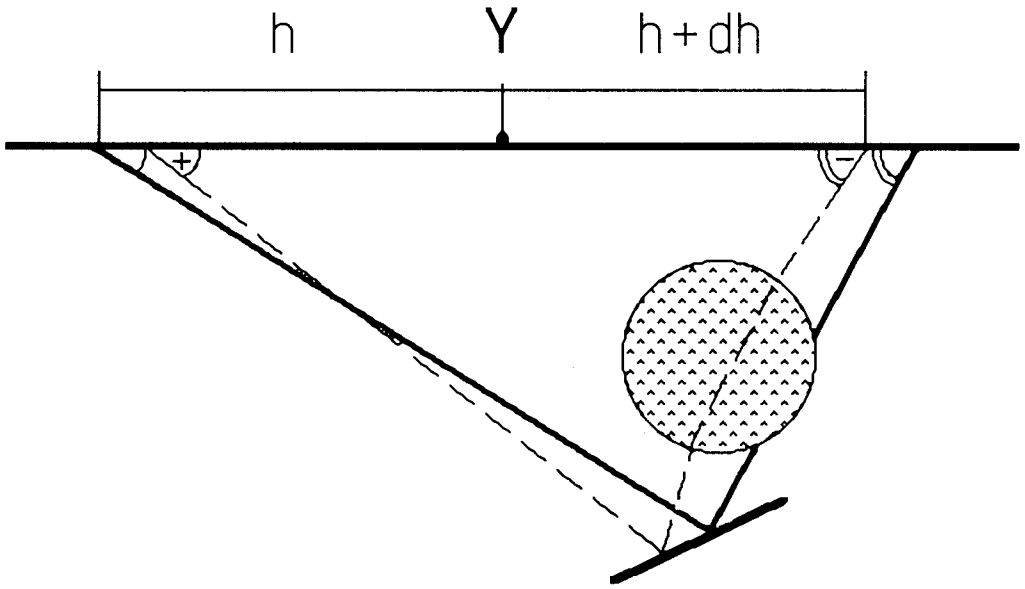

FIG. 2.12. The offset of the perturbed reflection (dashed line) is smaller than the offset of the unperturbed reflection (solid line) because the up-going ray passes below the center of the anomaly and deviates away from it.

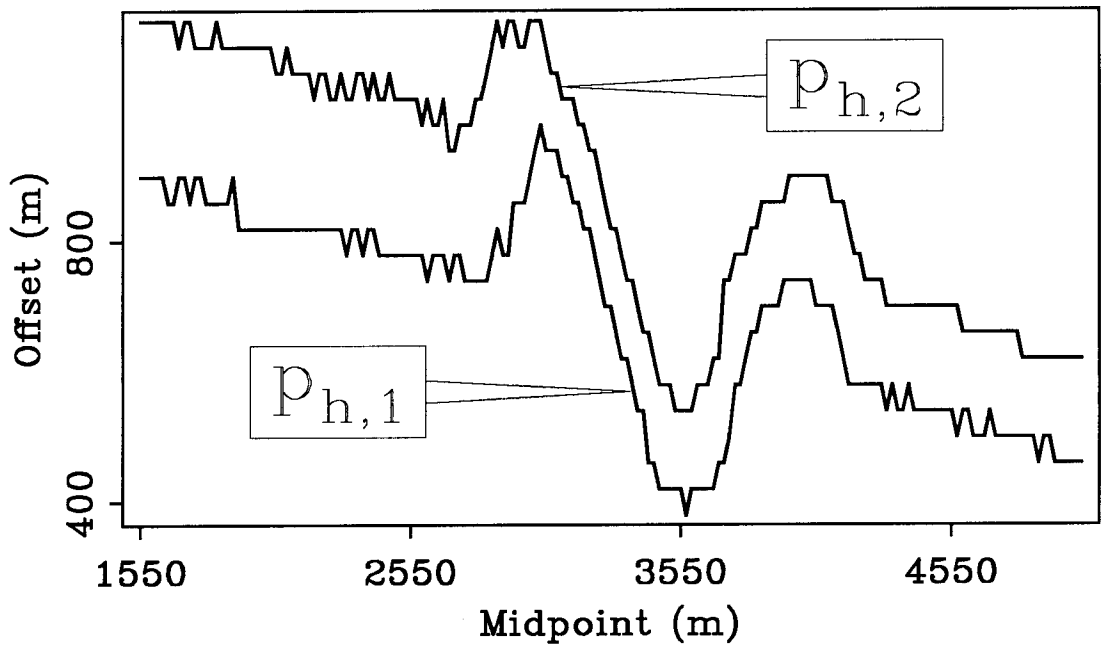

FIG. 2.13. The offsets of the beam-stacks' peaks as a function of midpoint location for two different offset ray parameters.

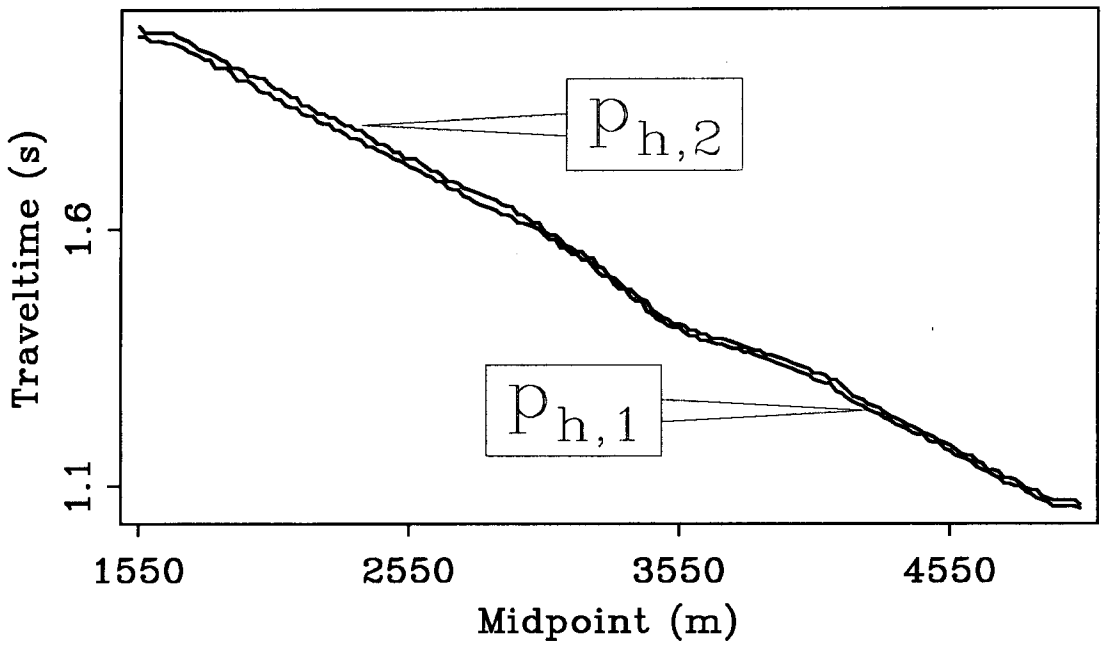

FIG. 2.14. The traveltimes of the beam-stacks' peaks as a function of midpoint location for two different offset ray parameters.

data. Therefore, for the estimation procedure to succeed, the ray tracing must correctly predict the offsets and traveltimes of the beam-stacks' peaks. Figure 2.15 compares the offsets predicted by ray tracing (solid line) and the offsets picked from the beam-stacked data (dotted line) for the offset ray parameter $p_{h,1}$. Figure 2.16 shows the same comparison for the traveltimes of beam-stacks. These two figures demonstrate that the behavior of beam-stacks can be well predicted with ray tracing.

The relative perturbations caused by the anomaly in the beam stacks' offsets (Figure 2.13 and Figure 2.15) are much larger than the relative perturbations in the beam stacks' traveltimes (Figure 2.14 and Figure 2.16). Beam stacks' offsets are considerably more sensitive to velocity perturbations than are traveltimes and thus they provide more reliable information on the velocity model than traveltimes do. Because of this effect the back projection operator that I will present in Chapter 3 relates perturbations in the velocity model to perturbations in the beam stacks' offsets, instead of relating the velocity perturbations to perturbations in the beam stacks' traveltimes.

The shape of the offset perturbations is similar to the effects of a velocity anomaly on stacking velocity (Toldi, 1985) and migration velocity (Fowler, 1988). The advantage in estimating velocity from beam stacks is that they make possible to measure the non-hyperbolic moveouts caused by the anomaly.

2.5 CONCLUSIONS

This chapter has introduced beam stack as a high-resolution local decomposition of the prestack data. Beams stack can be used for obtaining reliable information on the kinematics of the reflections even when lateral velocity variations cause non-hyperbolic moveouts.

Seismic velocity can be estimated by searching for the velocity model that best predicts the kinematics in the beam-stacked data. A good criterion for evaluating a velocity model is to sum beam stacks' semblance at the traveltimes and surface locations modeled by ray tracing. The maximization problem defined by this criterion is solved in the next chapter.

A synthetic example showed that beam stacks' offsets are considerably more sensitive to velocity perturbations than beam stacks' traveltimes are. This effect must be taken into account to define a stable velocity estimation algorithm.

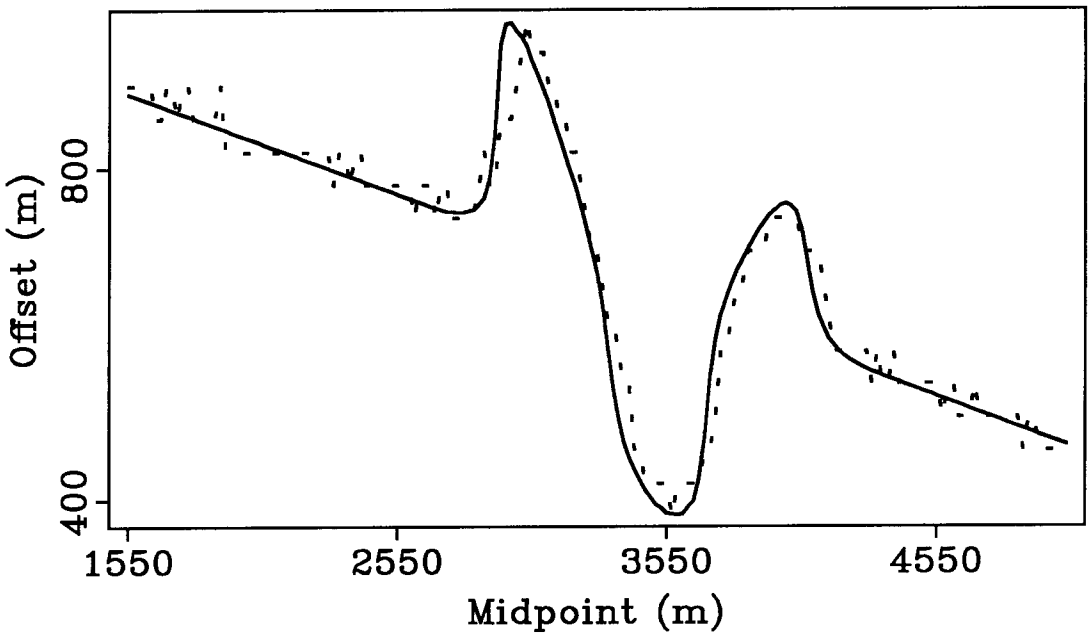

FIG. 2.15. The comparison of the offsets predicted by ray tracing (solid line) and those picked from beam-stacked data (dotted line).

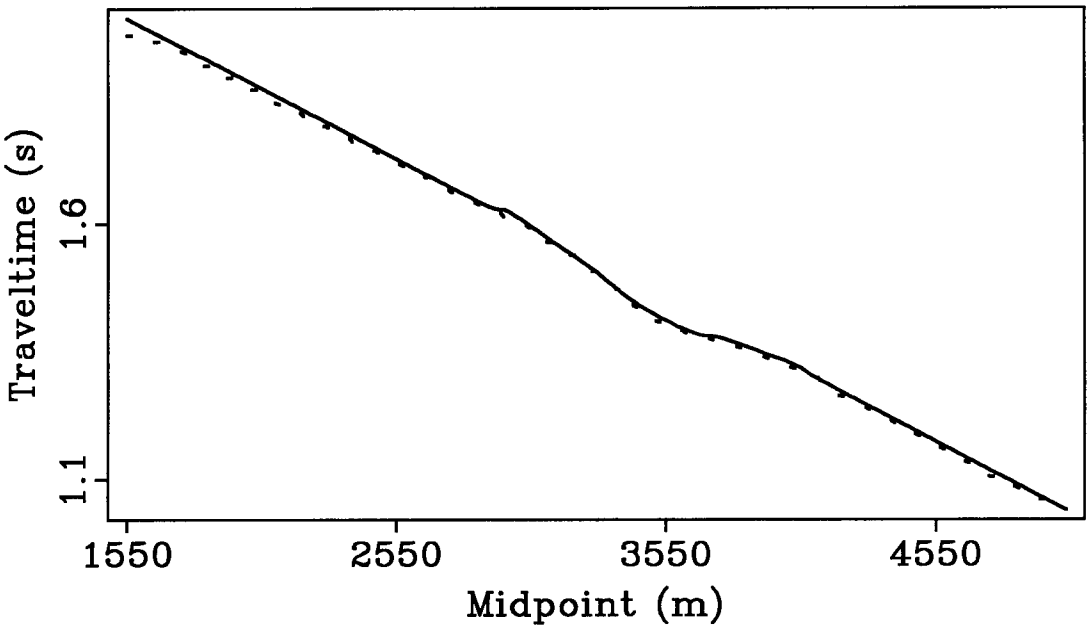

FIG. 2.16. The comparison of the traveltimes predicted by ray tracing (solid line) and those picked from beam-stacked data (dotted line).