

# Fast data extrapolating

Chunlin Wu, Jiansong Deng\*, Falai Chen

*Department of Mathematics, University of Science and Technology of China, Hefei, Anhui 230026, PR China*

Received 19 December 2005; received in revised form 28 May 2006

---

## Abstract

Data-extrapolating (extension) technique has important applications in image processing on implicit surfaces and in level set methods. The existing data-extrapolating techniques are inefficient because they are designed without concerning the specialities of the extrapolating equations. Besides, there exists little work on locating the narrow band after data extrapolating—a very important problem in narrow band level set methods. In this paper, we put forward the general Huygens' principle, and based on the principle we present two efficient data-extrapolating algorithms. The algorithms can easily locate the narrow band in data extrapolating. Furthermore, we propose a prediction–correction version for the data-extrapolating algorithms and the corresponding band locating method for a special case where the direct band locating method is hard to apply. Experiments demonstrate the efficiency of our algorithms and the convenience of the band locating method.

© 2006 Elsevier B.V. All rights reserved.

*MSC:* 65M99; 65D18; 68U05

*Keywords:* Data extrapolating; Huygens' principle; Image processing on surfaces; Level set methods

---

## 1. Introduction

Data extrapolating is an indispensable step in image processing on implicit surfaces and in level set methods. Processing images on surfaces is considered to be a very difficult problem by classical methods. The PDE-based methods developed in recent years spark a minor revolution in image processing and have become the principle methods [12,13,3,2,16,18] for processing images on surfaces. Implicit surfaces are a popular type of surfaces with enough flexibility in geometric modelling. Hence, processing images on implicit surfaces is an interesting and important problem. There exists a little work on processing images on implicit surfaces. Osher et al. studied the problem of denoising images on implicit surfaces [2], and the authors of the present paper proposed an inpainting algorithm for images on implicit surfaces [18]. The basis approach of these work is to set up some energy functionals, and by minimizing the energy functionals some PDEs are derived. The PDEs are then solved numerically in a narrow band near the given implicit surface, where image data extrapolating is needed.

Data extrapolating also plays an important role in level set methods [5,7,10]. In level set methods, one needs to know the velocities in the neighborhood of the interface (zero level set) in order to move the interface. In some applications such as mean curvature flow and constant normal flow, the velocities are given globally or at least near the interface.

---

\* Corresponding author. Tel.: +86 551 3601009; fax: +86 551 3601005.

*E-mail addresses:* [dengjs@ustc.edu.cn](mailto:dengjs@ustc.edu.cn) (J. Deng), [chenfl@ustc.edu.cn](mailto:chenfl@ustc.edu.cn) (F. Chen).

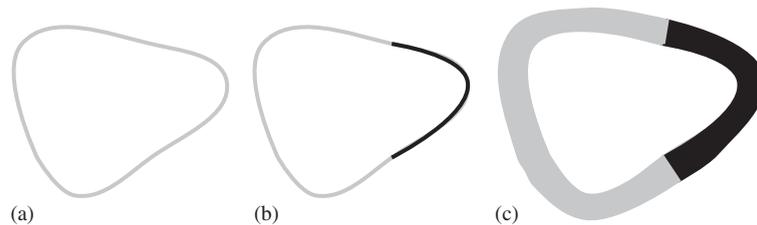


Fig. 1. Data extrapolating for an implicit curve: (a) an implicit curve, (b) data defined on the curve, and (c) the data are extrapolated in a band.

However, in most other applications such as the Stefan problem [5] and the Hele–Shaw flow problems [7], the velocities are given only at the interface. In these cases, one needs to extend the velocity to a band around the interface. After the data are extrapolated, one can then solve the corresponding level set equations in the band [10,15,4,6].

Data-extrapolating techniques are based on a Hamilton equation [10], which is also called the data-extrapolating equation. Assume an implicit surface  $S$  as the zero level set of  $\varphi$  is given. According to the data-extrapolating equation, data defined on  $S$  will be extended along the normals (outside of  $S$ ) or the opposites (inside of  $S$ ) of the level sets of  $\varphi$ . Fig. 1 illustrates an example of data extrapolating for an implicit curve. Here, a planar implicit curve is given in (a); in (b) a function (data) is defined on the curve, whose value is represented in gray on the left half while in black on the right half of the curve; (c) is the result of data extrapolating. The data defined on the curve are extrapolated to a band around the curve and the extrapolated data keep unchanged along the normal directions of the level sets of  $\varphi$ . Within the band around the curve as shown in Fig. 1(c), one can numerically solve the image processing equations or level set equations.

As far as the authors are aware, only a few data-extrapolating techniques are proposed so far. The first one is directly numerically solving the time-dependent data-extrapolating equation. According to its physical feature, a numerical method based on the upwind scheme can be adopted directly, see [10]. This method is simple in programming yet inefficient. If the data are extrapolated to the whole space, the algorithm complexity is  $O(N\sqrt[3]{N})$ , where  $N$  is the number of grids in space. Peng [11] proposed a fast local level set method where the equation is solved only in a narrow band near the interface. In each time step, the numerical method in [10] is directly adopted and all the grids are visited and processed. The algorithm is also simple but with an algorithm complexity  $O(N)$  since the data are extrapolated in a narrow band near the interface. Unfortunately, it is not reported on how to locate the narrow band in detail, and the band locating method seems to be useful only for signed distance implicit functions.

The second class of methods is just considering the steady state of the data-extrapolating equation and solving a corresponding time-independent equation. Adalsteinsson et al. [1] proposed a global data-extrapolating method based on heap-sorting. This method is similar to the fast marching method [14] and has a lower algorithm complexity  $O(N \log N)$ . But one should carefully design some complex data structures and implement a heap-sorting algorithm. As the steady-state equation of data extrapolating is a static Hamilton equation, fast sweeping methods based on Gauss–Seidel iteration can be adopted directly [9,17,8]. The algorithm complexity based on fast sweeping methods is  $O(N)$ . However, it is a global method and cannot be localized easily.

In this paper, we will present two new data-extrapolating algorithms based on the general Huygens' principle. The algorithms make full use of the physical properties of the time-dependent data-extrapolating equation. By our methods, the data are extrapolated to the whole space with only a complexity  $O(N)$ , and even with a lower complexity  $O(N^{2/3})$  if the data are extrapolated to only a narrow band. Our methods directly use the upwind scheme in each time step and do not involve any complex data structures. Thus, they are simple in programming. Unlike the local method in [11], only a small part of the grids needs to be processed in each time step by our methods. Furthermore, we can conveniently locate the narrow band for any implicit function according to the number of time steps in solving the data-extrapolating equation.

The paper is organized as follows. In Section 2, we analyze the data-extrapolating equation and present a method to locate the narrow band. The traditional data-extrapolating algorithm in [10,11] (for local case) is described briefly in Section 3. In Section 4, the general Huygens' principle is presented, and two fast data-extrapolating algorithms are proposed. In Section 5, for a special case where the locating of the narrow band is difficult, we put forward the prediction–correction version of the fast data-extrapolating methods. Experiments are provided in Section 6. Section 7 concludes the paper with a summary and future work.

### 2. Data-extrapolating equation

Assume an implicit surface  $S$  is given as the zero level set of  $\varphi : \mathbb{R}^3 \rightarrow \mathbb{R}$  with  $\varphi < 0$  representing the inside of  $S$  and  $\varphi > 0$  representing the outside of  $S$ . We assume  $|\nabla\varphi(x, y, z)| > 0$  for  $(x, y, z) \in S$ . For otherwise, if  $|\nabla\varphi(x, y, z)| = 0$  for some  $(x, y, z) \in S$ , we can reconstruct the surface  $S$  by its signed distance function such that  $|\nabla\varphi(x, y, z)| > 0$  holds for all  $(x, y, z) \in S$  [10]. It should be noticed that some implicit functions may not be differentiable. However, our discussion is still valid since the data-extrapolating algorithms are in discrete fashion. Let  $u$  be an image function (data) defined on  $S$ . The data-extrapolating equation is the following Hamilton equation [10]:

$$u_t + \text{sign}(\varphi)(\nabla u \cdot \nabla \varphi) = 0, \tag{1}$$

where  $\text{sign}(\varphi)$  is the sign function of  $\varphi$ . Based on this equation, data  $u$  defined on  $S$  will be extended along the normal directions (outside of  $S$ ) or the opposite normal directions (inside of  $S$ ) of the level sets of  $\varphi$ . Inside of  $S$ , data will be extrapolated to positions where the  $\varphi$  values become smaller; while outside of  $S$  data will be extrapolated to positions where  $\varphi$  values become larger. A band will be constructed after extrapolating. The larger the number of time steps of the extrapolating equation is solved, the wider the band will be. The following theorem describes the relationship between the width of the band and the number of time steps.

**Theorem 1.** *Under the evolution of Eq. (1), function  $u$  will be extrapolated along the normal directions (outside of  $S$ ) or the opposite normal directions (inside of  $S$ ) of the level sets of  $\varphi$ . At time  $t$ , the function will be extrapolated to  $X = \{(x(t), y(t), z(t)) : \varphi(x(t), y(t), z(t)) = \varphi_+(x(t), y(t), z(t))\}$  (outside of  $S$ ) and  $Y = \{(x(t), y(t), z(t)) : \varphi(x(t), y(t), z(t)) = \varphi_-(x(t), y(t), z(t))\}$  (inside of  $S$ ), where*

$$\begin{aligned} \varphi_+(x(t), y(t), z(t)) &= \int_0^t |\nabla\varphi(x_+(\tau), y_+(\tau), z_+(\tau))|^2 d\tau, \\ \varphi_-(x(t), y(t), z(t)) &= - \int_0^t |\nabla\varphi(x_-(\tau), y_-(\tau), z_-(\tau))|^2 d\tau. \end{aligned} \tag{2}$$

Especially we have the following estimations:

1.  $u$  will be extrapolated to the level sets  $\tilde{X} = \{(x, y, z) : \varphi(x, y, z) = \tilde{\varphi}_+\}$  (outside of  $S$ ) and  $\tilde{Y} = \{(x, y, z) : \varphi(x, y, z) = \tilde{\varphi}_-\}$  (inside of  $S$ ), where

$$\begin{aligned} \tilde{\varphi}_+ &= \min_{(x,y,z), \varphi(x,y,z) \geq 0} |\nabla\varphi|^2 t = t \text{Min } G_+, \\ \tilde{\varphi}_- &= - \min_{(x,y,z), \varphi(x,y,z) \leq 0} |\nabla\varphi|^2 t = -t \text{Min } G_-. \end{aligned} \tag{3}$$

2.  $u$  will at most be extrapolated to the level sets  $\hat{X} = \{(x, y, z) : \varphi(x, y, z) = \hat{\varphi}_+\}$  (outside of  $S$ ) and  $\hat{Y} = \{(x, y, z) : \varphi(x, y, z) = \hat{\varphi}_-\}$  (inside of  $S$ ), where

$$\begin{aligned} \hat{\varphi}_+ &= \max_{(x,y,z), \varphi(x,y,z) \geq 0} |\nabla\varphi|^2 t = t \text{Max } G_+, \\ \hat{\varphi}_- &= - \max_{(x,y,z), \varphi(x,y,z) \leq 0} |\nabla\varphi|^2 t = -t \text{Max } G_-. \end{aligned} \tag{4}$$

**Proof.** We just prove the case for the outside of  $S$ . The information of  $u$  at any point  $\mathbf{p}_s \in S$  will march along a normal curve  $C_n$  with starting point  $\mathbf{p}_s$ . Here a “normal curve” is a curve getting started from a point on surface  $S$  and at any point  $\mathbf{p}$  on the curve, the tangent vector of the curve at  $\mathbf{p}$  is the normal vector of the level sets of  $\varphi$  through  $\mathbf{p}$ . At time  $t$ , the object (function information) at  $\mathbf{p}_s \in S$  will arrive at a point  $\mathbf{p}_t = (x(t), y(t), z(t)) \in X$ , and the function  $u$  is extended to  $X = \{(x(t), y(t), z(t)) : \varphi(x(t), y(t), z(t)) = \varphi_+(x(t), y(t), z(t))\}$ . The extrapolating velocity at  $\mathbf{p}_t$  is

$$\frac{dx}{dt} = \varphi_x(x(t), y(t), z(t)), \quad \frac{dy}{dt} = \varphi_y(x(t), y(t), z(t)), \quad \frac{dz}{dt} = \varphi_z(x(t), y(t), z(t)).$$

Thus

$$x(t) = \int_0^t \varphi_x(x(\tau), y(\tau), z(\tau)) \, d\tau + x_s,$$

$$y(t) = \int_0^t \varphi_y(x(\tau), y(\tau), z(\tau)) \, d\tau + y_s,$$

$$z(t) = \int_0^t \varphi_z(x(\tau), y(\tau), z(\tau)) \, d\tau + z_s.$$

Therefore,

$$\varphi_+(x(t), y(t), z(t)) = \varphi \left( x_s + \int_0^t \varphi_x(x(\tau), y(\tau), z(\tau)) \, d\tau, y_s + \int_0^t \varphi_y(x(\tau), y(\tau), z(\tau)) \, d\tau, z_s + \int_0^t \varphi_z(x(\tau), y(\tau), z(\tau)) \, d\tau \right).$$

Differentiating both sides of the above equation with respect to  $t$ , we obtain

$$\frac{d\varphi_+}{dt} = \varphi_x \varphi_x + \varphi_y \varphi_y + \varphi_z \varphi_z = |\nabla \varphi(x(t), y(t), z(t))|^2.$$

Integrating this equation and noticing that the data are extrapolated from the surface  $S(\varphi = 0)$ , the first part of the theorem is proved. Since

$$\min_{(x,y,z), \varphi(x,y,z) \geq 0} |\nabla \varphi|^2 t \leq \varphi_+ \leq \max_{(x,y,z), \varphi(x,y,z) \geq 0} |\nabla \varphi|^2 t,$$

$$-\max_{(x,y,z), \varphi(x,y,z) \leq 0} |\nabla \varphi|^2 t \leq \varphi_- \leq -\min_{(x,y,z), \varphi(x,y,z) \leq 0} |\nabla \varphi|^2 t,$$

the estimations hold. Notations  $\text{Min } G_+$  and  $\text{Min } G_-$  in (3),  $\text{Max } G_+$  and  $\text{Max } G_-$  in (4) are introduced for simplicity. □

Because the norms of the normal vectors at different positions along level sets vary, the extrapolating velocities of data on  $S$  vary as well. Data will be extrapolated fast in some directions while slowly in other directions. Our first estimation points out that all data will be extended to  $\hat{X}$  and  $\hat{Y}$  at time  $t$ . The second estimation tells that the data with the fastest extrapolating velocity could be extended to  $\hat{X}$  and  $\hat{Y}$ . The first estimation provides a method for locating a narrow band inside which image processing equations and/or level set equations can be solved. These two estimations will help us to construct the fast data-extrapolating algorithms.

The numerical scheme for Eq. (1) consists of discretizations of  $\varphi$  and  $u$ . A numerical method should consider the physical properties of the data-extrapolating equation. Data on  $S$  is extended along the normal directions (outside of  $S$ ) and the opposite normal directions (inside of  $S$ ) of the level sets of  $\varphi$ . Physically we can regard this extending procedure as a flow and the data on  $S$  as the inflow condition. In our problem, inflow keeps unchanged. Outside of  $S$  the data continuously flow from positions with smaller  $\varphi$  values to positions with larger  $\varphi$  values; while inside of  $S$  the flow direction is opposite. According to this observation, we can easily write down the discretizations of  $\varphi$  and  $u$  [10]. We assume the spacial division is  $\{(i \Delta x, j \Delta y, k \Delta z), 1 \leq i \leq N_x, 1 \leq j \leq N_y, 1 \leq k \leq N_z\}$ . Let  $\varphi_{i,j,k} = \varphi(i \Delta x, j \Delta y, k \Delta z)$ ,  $u_{i,j,k} = u(i \Delta x, j \Delta y, k \Delta z)$ .

At first, we discretize  $\nabla \varphi$  in Eq. (1). As an example, just  $\varphi_x$  is discretized. The discretizations of  $\varphi_y$  and  $\varphi_z$  are similar. Outside of  $S$ , if  $\varphi_{i+1,j,k} > \varphi_{i,j,k}$  and  $\varphi_{i-1,j,k} > \varphi_{i,j,k}$ , then  $(\varphi_x)_{i,j,k} = 0$ ; otherwise, if  $\varphi_{i-1,j,k} < \varphi_{i+1,j,k}$ , then  $(\varphi_x)_{i,j,k} = (\varphi_{i,j,k} - \varphi_{i-1,j,k})/\Delta x$ ; else if  $\varphi_{i-1,j,k} \geq \varphi_{i+1,j,k}$ , then  $(\varphi_x)_{i,j,k} = (\varphi_{i+1,j,k} - \varphi_{i,j,k})/\Delta x$ . Inside of  $S$ , if  $\varphi_{i+1,j,k} < \varphi_{i,j,k}$  and  $\varphi_{i-1,j,k} < \varphi_{i,j,k}$ , then  $(\varphi_x)_{i,j,k} = 0$ ; otherwise, if  $\varphi_{i-1,j,k} < \varphi_{i+1,j,k}$ , then  $(\varphi_x)_{i,j,k} = (\varphi_{i+1,j,k} - \varphi_{i,j,k})/\Delta x$ ; else if  $\varphi_{i-1,j,k} \geq \varphi_{i+1,j,k}$ , then  $(\varphi_x)_{i,j,k} = (\varphi_{i,j,k} - \varphi_{i-1,j,k})/\Delta x$ . On  $S$ , because the data keep unchanged,  $(\varphi_x)_{i,j,k} = 0$ .  $\nabla \varphi$  is fixed with respect to time  $t$ , so it can be computed only once.

An upwind scheme coupled with a forward Euler time discretization is directly adopted for  $u$ .

### 3. Traditional data-extrapolating algorithm

Assume that a time step size  $\Delta t$  is chosen under CFL condition and Eq. (1) is solved with  $N_T$  time steps. The traditional data-extrapolating algorithm can be described in pseudo-code as follows.

---

**Algorithm 1** (Traditional data-extrapolating algorithm).

---

```

Begin
  for  $n = 1$  to  $N_T$ 
    for  $i = 1$  to  $N_X$ 
      for  $j = 1$  to  $N_Y$ 
        for  $k = 1$  to  $N_Z$ 
          Process a grid point;
        for  $i = 1$  to  $N_X$ 
          for  $j = 1$  to  $N_Y$ 
            for  $k = 1$  to  $N_Z$ 
              Update a grid point.

```

---

In this algorithm, all grids will be visited in each time step. Hence, its complexity is very high for multi-dimension problems. According to our experiments, data extrapolating in this fashion costs much more time than solving other equations such as level set equations. Although in the local level set method the extension has only complexity  $O(N)$  [11], it is also necessary that all the grids are visited and processed in each time step. In the next section, we will present a general Huygens' principle according to Theorem 1 and then propose two fast algorithms for data extrapolating. In our algorithms, only those grids near a surface are processed in each time step. Intrinsically, 3D problems are converted to 2D problems and the number of processed grids in each time step is reduced dramatically.

### 4. Fast data-extrapolating algorithms

Data-extrapolating equation (1) is essentially a wave equation. Assume the equation is solved till some time  $t$ . Starting from this moment  $t$ , those grids to which the data have been extended become new data sources (field sources or wave sources). This is the so-called *general Huygens' principle*. Precisely, we have

**Principle 1.** *At time  $t = 0$  the data defined on the zero level set of  $\varphi$  (surface  $S$ ) are field sources. Under the evolution of Eq. (1), the data are continuously extrapolated. According to Theorem 1, at time  $t$  the data are extrapolated to  $X$  and  $Y$ . For the next moment, the data on  $X$  and  $Y$  become new sources. The data will be kept on extrapolating from these new sources as  $t$  increases. If the data on  $S$  (initial wave sources) keep unchanged as time varies, we can regard  $\tilde{X} = \{(x, y, z) : \varphi(x, y, z) = t \text{ Min } G_+\}$  (outside of  $S$ ) and  $\tilde{Y} = \{(x, y, z) : \varphi(x, y, z) = -t \text{ Min } G_-\}$  (inside of  $S$ ) as new wave sources at time  $t$ .*

Here the term “general” comes from three aspects:

1. In the classical Huygens' principle, only points are considered to be sources in space. While in the general Huygens' principle sources are implicit surfaces.
2. In the classical Huygens' principle, the wave speed is isotropic and the magnitude of the wave speed is a constant. But it is anisotropic in the general Huygens' principle depending on the normals of the level sets of  $\varphi$ .
3. The locating of wave front is implicit in the general Huygens' principle using the  $\varphi$  values, while explicit in the classical Huygens' principle using the geometric positions of the front. The geometric positions can be implicitly represented by a special function called signed distance function.

Based on the general Huygens' principle, data extrapolating becomes simple because the inflow (data, or wave sources) keeps invariant with time changes and the procedure of data extrapolating is in fact a field-generating procedure. Our strategy is to skip those grids in each time step to which the data are impossible to be extended or the data

have been extended before this time step. Specifically, according to Theorem 1, at time  $t$ , the data are extended to  $\tilde{X}_t = \{(x, y, z) : \varphi(x, y, z) = t \text{ Min } G_+\}$  (outside of  $S$ ) and  $\tilde{Y}_t = \{(x, y, z) : \varphi(x, y, z) = -t \text{ Min } G_-\}$  (inside of  $S$ ). From  $t$  to  $t + \Delta t$ , the data will be at most extrapolated to  $\tilde{X}_{t+\Delta t} = \{(x, y, z) : \varphi(x, y, z) = (t + \Delta t) \text{ Max } G_+\}$  (outside of  $S$ ) and  $\tilde{Y}_{t+\Delta t} = \{(x, y, z) : \varphi(x, y, z) = -(t + \Delta t) \text{ Max } G_-\}$  (inside of  $S$ ). Therefore, in the time step from  $t$  to  $t + \Delta t$ , we only have to solve the extrapolating equation in the band  $\{(x, y, z) : t \text{ Min } G_+ \leq \varphi(x, y, z) \leq (t + \Delta t) \text{ Max } G_+\}$  and  $\{(x, y, z) : -(t + \Delta t) \text{ Max } G_- \leq \varphi(x, y, z) \leq -t \text{ Min } G_-\}$ . It should be pointed out that some grids could be missed (not be processed) from the above criteria. We modify the criteria by checking the maximal change of  $\varphi$  in a grid cell to avoid this phenomenon. According to the Lagrange mean-value theorem and the Schwartz inequality, the maximal change of  $\varphi$  in a grid cell does not exceed  $\text{Max } G_+ \sqrt{(\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2}$  (outside of  $S$ ) and  $\text{Max } G_- \sqrt{(\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2}$  (inside of  $S$ ). Taking this into consideration, we get our first data-extrapolating algorithm which is described in Algorithm 2. As one can see that only the grids in a narrow band need to be processed in each time step.

---

**Algorithm 2** (Fast data-extrapolating algorithm I).

---

Begin

```

 $t = 0, L = \sqrt{(\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2}$ 
for  $n = 1$  to  $N_T$ 
  for  $i = 1$  to  $N_X$ 
    for  $j = 1$  to  $N_Y$ 
      for  $k = 1$  to  $N_Z$ 
        if  $t \text{ Min } G_+ - L \text{ Max } G_+ \leq \varphi_{i,j,k} \leq (t + \Delta t) \text{ Max } G_+ + L \text{ Max } G_+$  or
            $-(t + \Delta t) \text{ Max } G_- - L \text{ Max } G_- \leq \varphi_{i,j,k} \leq -t \text{ Min } G_- + L \text{ Max } G_-$ 
          Process a grid point;
      for  $i = 1$  to  $N_X$ 
        for  $j = 1$  to  $N_Y$ 
          for  $k = 1$  to  $N_Z$ 
            if  $t \text{ Min } G_+ - L \text{ Max } G_+ \leq \varphi_{i,j,k} \leq (t + \Delta t) \text{ Max } G_+ + L \text{ Max } G_+$  or
                $-(t + \Delta t) \text{ Max } G_- - L \text{ Max } G_- \leq \varphi_{i,j,k} \leq -t \text{ Min } G_- + L \text{ Max } G_-$ 
              Update a grid point;
           $t = t + \Delta t$ 

```

End

---

The above algorithm can be improved as follows. Since image processing equations or level set equations are solved in a narrow band  $B = \{(x, y, z) : -T \text{ Min } G_- \leq \varphi(x, y, z) \leq T \text{ Min } G_+\}$ , where  $T > 0$ , it is only necessary to consider those grids determined by the first estimation of Theorem 1 in each time step. From time  $t$  to  $t + \Delta t$ , the data will be extrapolated to  $\tilde{X}_{t+\Delta t} = \{(x, y, z) : \varphi(x, y, z) = (t + \Delta t) \text{ Min } G_+\}$  (outside of  $S$ ) and  $\tilde{Y}_{t+\Delta t} = \{(x, y, z) : \varphi(x, y, z) = -(t + \Delta t) \text{ Min } G_-\}$  (inside of  $S$ ). Therefore, in the time step from  $t$  to  $t + \Delta t$ , only those grids satisfying  $t \text{ Min } G_+ \leq \varphi_{x,y,z} \leq (t + \Delta t) \text{ Min } G_+$  or  $-(t + \Delta t) \text{ Min } G_- \leq \varphi_{x,y,z} \leq -t \text{ Min } G_-$  need to be processed. Based on the above analysis, we deduce the second data-extrapolating algorithm which is described in Algorithm 3.

---

**Algorithm 3** (Fast data-extrapolating algorithm II).

---

Begin

```

 $t = 0, L = \sqrt{(\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2}$ 
for  $n = 1$  to  $N_T$ 
  for  $i = 1$  to  $N_X$ 
    for  $j = 1$  to  $N_Y$ 
      for  $k = 1$  to  $N_Z$ 
        if  $t \text{ Min } G_+ - \text{Max } G_+ L \leq \varphi_{i,j,k} \leq (t + \Delta t) \text{ Min } G_+ + \text{Max } G_+ L$  or
            $-(t + \Delta t) \text{ Min } G_- - \text{Max } G_- L \leq \varphi_{i,j,k} \leq -t \text{ Min } G_- + \text{Max } G_- L$ 
          Process a grid point;

```

```

for i = 1 to NX
for j = 1 to NY
for k = 1 to NZ
    if t Min G+ - Max G+L ≤ φi,j,k ≤ (t + Δt) Min G+ + Max G+L or
    -(t + Δt) Min G- - Max G-L ≤ φi,j,k ≤ -t Min G- + Max G-L
        Update a grid point;
t = t + Δt

```

End

### 5. Data extrapolating based on prediction–correction

Based on Theorem 1 and the general Huygens’ principle, we have proposed two fast data-extrapolating algorithms. But there is a special case in which our fast data-extrapolating algorithms do not work efficiently and the determination of the narrow band is difficult. That is, if  $\text{Min } G_+ = 0$  or  $\text{Min } G_- = 0$ , the locating of the narrow band fails and algorithm complexity increases greatly. Here, we propose the improved versions of the algorithms based on prediction–correction.

Because of the assumption that  $|\nabla\varphi(x, y, z)| > 0$  for any  $(x, y, z) \in S$ , we have  $|\nabla\varphi(x, y, z)| > 0$  when  $-t \text{Max } G_- \leq \varphi(x, y, z) \leq t \text{Max } G_+$  for some sufficiently small  $t$ . Now suppose the data are extrapolated till some time  $t$  by solving Eq. (1), the data are at most extended to the level sets  $\varphi = t \text{Max } G_+$  (outside of  $S$ ) and  $\varphi = -t \text{Max } G_-$  (inside of  $S$ ). This is the prediction step. By the same reason as described in Algorithm A2 and 3, the extrapolating procedure will not change when we solve the extrapolating equation (1) in the band  $\{(x, y, z) : -t \text{Max } G_- \leq \varphi(x, y, z) \leq t \text{Max } G_+\}$ . Applying Theorem 1 again in the above band, we know that the data are at least extended to the level sets  $\varphi = t \text{Min } G'_+$  (outside of  $S$ ) and  $\varphi = -t \text{Min } G'_-$  (inside of  $S$ ), where

$$\text{Min } G'_+ = \min_{\substack{(x,y,z) \\ 0 \leq \varphi(x,y,z) \leq t \text{Max } G_+}} |\nabla\varphi|^2, \quad \text{Min } G'_- = \min_{\substack{(x,y,z) \\ -t \text{Max } G_- \leq \varphi(x,y,z) \leq 0}} |\nabla\varphi|^2.$$

And the data are at most extended to the level sets  $\varphi = t \text{Max } G'_+$  (outside of  $S$ ) and  $\varphi = -t \text{Max } G'_-$  (inside of  $S$ ), where

$$\text{Max } G'_+ = \max_{\substack{(x,y,z) \\ 0 \leq \varphi(x,y,z) \leq t \text{Max } G_+}} |\nabla\varphi|^2, \quad \text{Max } G'_- = \max_{\substack{(x,y,z) \\ -t \text{Max } G_- \leq \varphi(x,y,z) \leq 0}} |\nabla\varphi|^2.$$

This is the correction step. Replacing  $\text{Max } G_+$ ,  $\text{Max } G_-$ ,  $\text{Min } G_+$  and  $\text{Min } G_-$  by  $\text{Max } G'_+$ ,  $\text{Max } G'_-$ ,  $\text{Min } G'_+$  and  $\text{Min } G'_-$ , we can get the prediction–correction version of Theorem 1 on data extrapolating. The corresponding general Huygens’ principle and fast data-extrapolating algorithms can be also obtained. The details are omitted.

It should be pointed out that the prediction–correction based algorithms dramatically increase algorithm efficiency for the case where  $\text{Max } G_+$  or  $\text{Max } G_-$  is very large. Obviously we have

$$\text{Max } G'_+ \leq \text{Max } G_+, \text{Max } G'_- \leq \text{Max } G_-, \text{Min } G'_+ \geq \text{Min } G_+, \text{Min } G'_- \geq \text{Min } G_-.$$

Therefore, algorithms based on prediction–correction behave better and the locating of narrow band is more precise in many situations.

### 6. Examples

Three examples are provided to demonstrate the efficiency of our data-extrapolating algorithms. In all the examples, the extrapolated data by the traditional data-extrapolating algorithm and two data-extrapolating algorithms presented

Table 1  
CPU time

	Vase with strips	Horse with a character	Lena on a sphere
Traditional algorithm	2 min 23 s	17 min 20 s	3 min 15 s
Algorithm I	1 min 32 s	12 min 55 s	2 min 18 s
Algorithm II	1 min 12 s	6 min 33 s	1 min 43 s

in the paper (algorithms I and II) are illustrated and their CPU time is listed in Table 1. The experiments are performed on a PC with Pentium III 2.0 GHz CPU and 512 MB memory under Windows XP system.

In the first example, we are given a vase with five circular black strips as shown in Fig. 2. We compute

$$\text{Max } G_+ = 1.26121, \quad \text{Min } G_+ = 0.781363, \quad \text{Max } G_- = 1.09372, \quad \text{Min } G_- = 0.462234.$$

After 200 time steps, data are extrapolated to the level sets  $\varphi = -0.0169652$  (inside of the vase) and  $\varphi = 0.0368462$  (outside of the vase).

The next example illustrates a horse with a Chinese character on the horse (as shown in Fig. 3). One computes

$$\text{Max } G_+ = 1.45384, \quad \text{Min } G_+ = 0.178666, \quad \text{Max } G_- = 1.25715, \quad \text{Min } G_- = 0.20835.$$

After 900 time steps, the data are extrapolated to the level sets  $\varphi = -0.00983553$  (inside of the horse) and  $\varphi = 0.00723261$  (outside of the horse).

The final example is Lena on a sphere as shown in Fig. 4. In this example, data-extrapolating algorithms based on prediction–correction are applied because  $\text{Min } G_- = 0$ . The prediction step is achieved by solving the extrapolating equation in 300 time steps. In the correction step, we locate the narrow band more precisely and obtain

$$\text{Max } G'_+ = 1.83833, \quad \text{Min } G'_+ = 1.0068, \quad \text{Max } G'_- = 0.992844, \quad \text{Min } G'_- = 0.705791.$$

After 300 time steps, data are extrapolated to the level sets  $\varphi = -0.065098$  (inside of the sphere) and  $\varphi = 0.132465$  (outside of the sphere). Fig. 4 shows the result.

From Table 1, we can see that the data-extrapolating algorithms proposed in the paper are more efficient than the traditional data-extrapolating algorithm.

## 7. Conclusion

In this paper, we present two data-extrapolating algorithms based on the general Huygens' principle. The algorithms are based on solving a Hamilton equation in the neighborhood of an implicitly defined surface. In each time step, only the data at a small portion of grids need to be computed and updated, and thus the algorithm complexities are dramatically reduced. Furthermore, a method is proposed for implicitly locating a narrow band. For a special case where the data-extrapolating algorithms and the band locating method fail, we improve the algorithms and the method by a prediction–correction approach. Our algorithms are simple to implement and have much lower algorithm complexities. Examples demonstrate the efficiency of our algorithms and preciseness of the band locating method.

The future work is to study fast dynamic data-extrapolating algorithms. Notice that  $\text{Max } G_+$ ,  $\text{Max } G_-$ ,  $\text{Min } G_+$ , and  $\text{Min } G_-$  can be modified dynamically and locally according to the positions where the data are extrapolated. Therefore, dynamic algorithms with more effectiveness and more preciseness of band locating could be designed. However, it is also expensive to compute dynamic values  $\text{Max } G_+$ ,  $\text{Max } G_-$ ,  $\text{Min } G_+$  and  $\text{Min } G_-$ . How to balance between the algorithm complexity and the addition cost to compute the dynamic values  $\text{Max } G_+$ ,  $\text{Max } G_-$ ,  $\text{Min } G_+$  and  $\text{Min } G_-$  is the main issue to be solved.

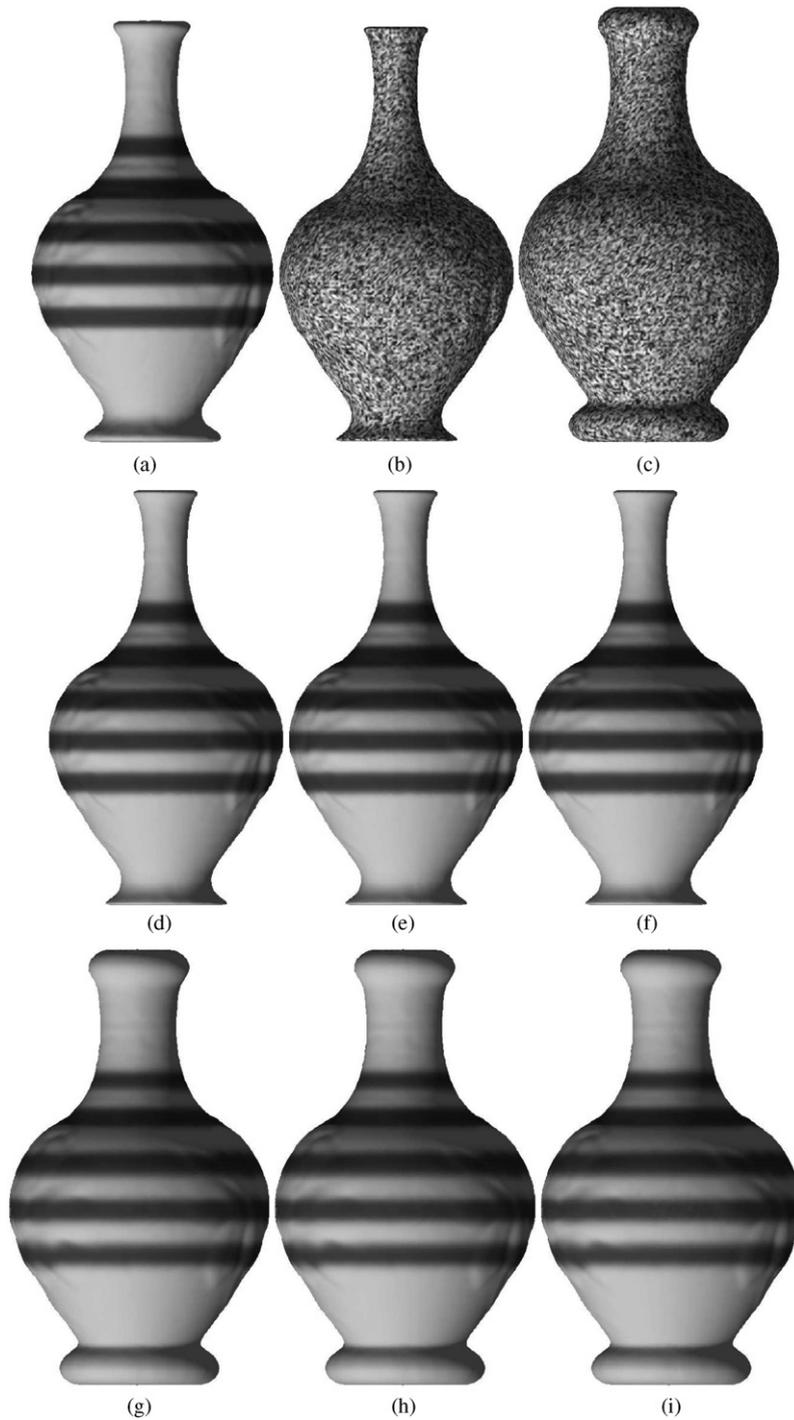


Fig. 2. Strips on a vase: (a–c) are the data (images) defined on the level sets  $\varphi = 0.0$ ,  $-0.0169652$  and  $0.0368462$  before extrapolating; (d–f) are the data defined on the level set  $\varphi = -0.0169652$  (inside of the surface) using the traditional data-extrapolating algorithm and the fast data-extrapolating algorithms I and II; (g–i) are the data defined on the level set  $\varphi = 0.0368462$  (outside of the surface) using the traditional data-extrapolating algorithm and the fast data-extrapolating algorithms I and II.

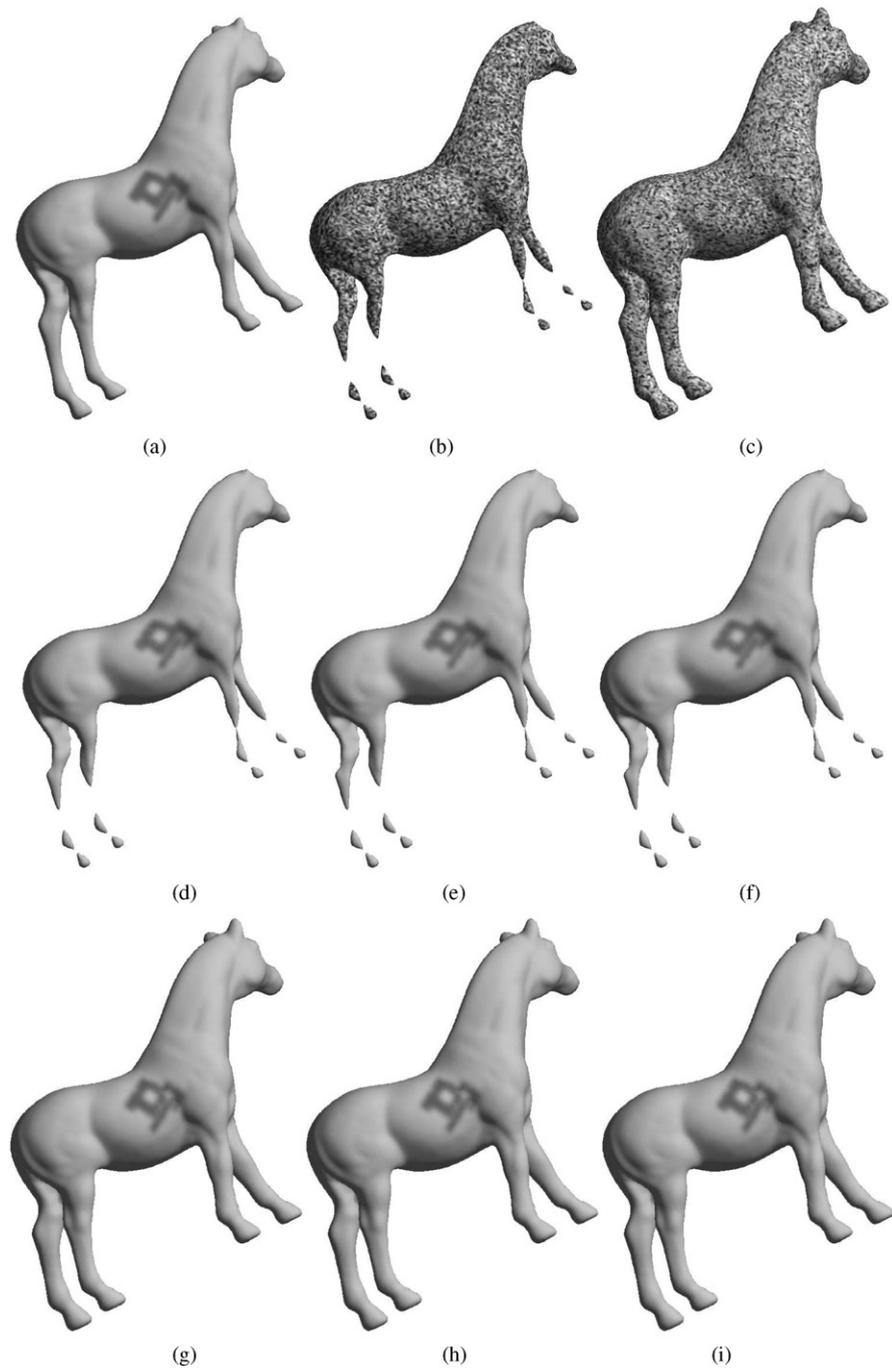


Fig. 3. A horse with a Chinese character: (a–c) are the data (images) defined on the level sets  $\varphi = 0.0$ ,  $-0.00983553$  and  $0.00723261$  before extrapolating; (d–f) are the data defined on the level set  $\varphi = -0.00983553$  (inside of the surface) using the traditional data-extrapolating algorithm and the fast data-extrapolating algorithms I and II; (g–i) are the data defined on the level set  $\varphi = 0.00723261$  (outside of the surface) using the traditional data-extrapolating algorithm and the fast data-extrapolating algorithms I and II.

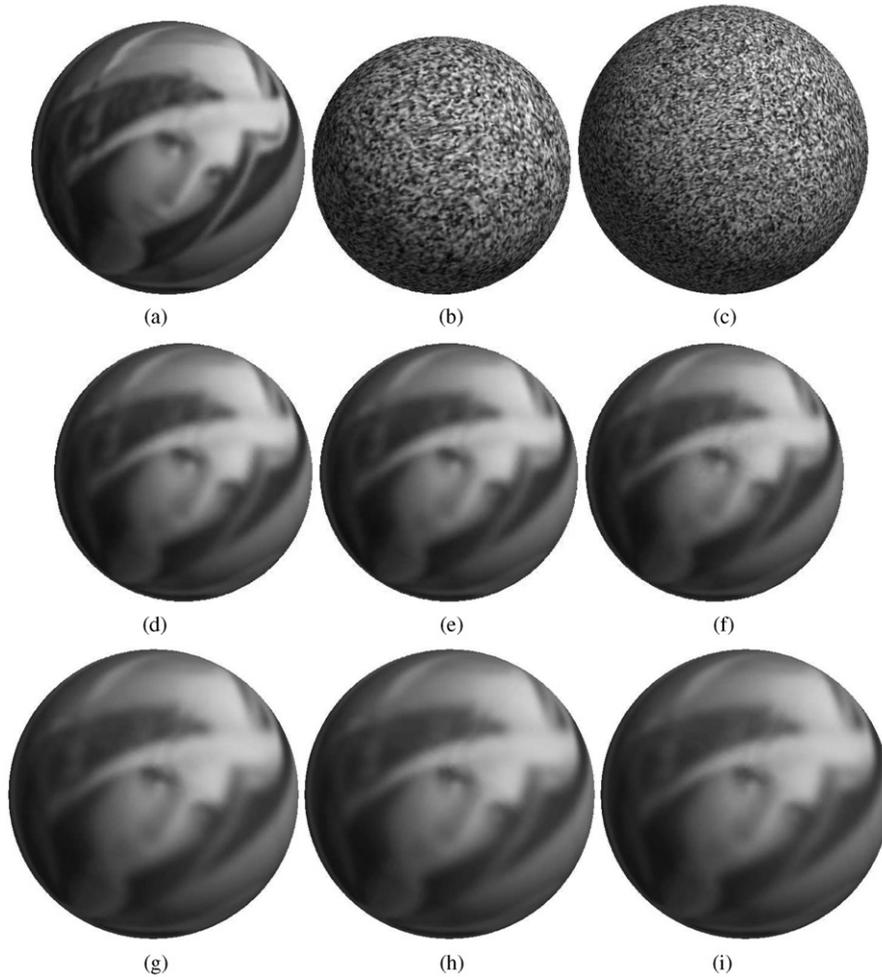


Fig. 4. Lena on a sphere: (a–c) are the data (images) defined on level sets  $\varphi = 0.0$ ,  $-0.065098$  and  $0.132465$  before extrapolating; (d–f) are the data defined on the level set  $\varphi = -0.065098$  (inside of the surface) using the traditional data-extrapolating algorithm and the fast data-extrapolating algorithms I and II based on prediction–correction; (g–i) are the data defined on the level set  $\varphi = 0.132465$  (outside of the surface) using the traditional data-extrapolating algorithm and the fast data-extrapolating algorithms I and II based on prediction–correction.

## Acknowledgment

The authors are supported by the Outstanding Youth Grant of NSF of China (no. 60225002), a National Key Basic Research Project of China (2004CB318000) and NSF of China (Nos. 60533060 and 60473132).

## References

- [1] D. Adalsteinsson, J.A. Sethian, The fast construction of extension velocities in level set methods, *J. Comput. Phys.* 148 (1999) 2–22.
- [2] M. Bertalmio, L.T. Cheng, S. Osher, G. Sapiro, Variational problems and partial differential equations on implicit surfaces, *J. Comput. Phys.* 174 (2) (2001) 759–780.
- [3] T.F. Chan, J. Shen, L. Vese, Variational PDE models in image processing, *Notices Amer. Math. Soc.* 50 (2003) 14–26.
- [4] S. Chen, B. Merriman, M. Kang, R.E. Caffisch, C. Ratsch, L.T. Cheng, M. Gyure, R.P. Fedkiw, C. Anderson, S. Osher, A level set method for thin film epitaxial growth, *J. Comput. Phys.* 167 (2001) 475–500.
- [5] S. Chen, B. Merriman, S. Osher, P. Smereka, A simple level set method for solving Stefan problems, *J. Comput. Phys.* 135 (1997) 8–29.
- [6] L.T. Cheng, P. Burchard, B. Merriman, S. Osher, Motion of curves constrained on surfaces using a level set approach, *J. Comput. Phys.* 175 (2002) 604–644.

- [7] T. Hou, Z. Li, S. Osher, H.K. Zhao, A hybrid method for moving interface problems with applications to the Hele–Shaw flows, *J. Comput. Phys.* 134 (1997) 236–252.
- [8] C.Y. Kao, S.J. Osher, J. Qian, Lax–Friedrichs sweeping schemes for static Hamilton–Jacobi equations, *J. Comput. Phys.* 196 (2004) 367–391.
- [9] S. Leung, J. Qian, An adjoint state method for 3-D transmission traveltime tomography using first-arrivals, *Comm. Math. Sci.* 4 (2006) 249–266 to appear.
- [10] S. Osher, R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, Springer, Berlin, 2002.
- [11] D. Peng, B. Merriman, S. Osher, H.K. Zhao, M.Kang, A PDE based fast local level set method, *J. Comput. Phys.* 155 (1999) 410–438.
- [12] P. Perona, J. Malik, Scale-space and edge detection using anisotropic diffusion, *IEEE Trans. Pattern Anal. Machine Intelligence* 12 (7) (1990) 629–639.
- [13] L. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, *Physica D* 60 (1992) 259–268.
- [14] J.A. Sethian, A fast marching level set method for monotonically advancing fronts, *Proc. Nat. Acad. Sci.* 93 (4) (1996) 1591–1595.
- [15] P. Smereka, Semi-implicit level set methods for curvature and surface diffusion motion, *J. Sci. Comput.* 19 (1–3) (2003) 439–456.
- [16] B. Tang, G. Sapiro, V. Caselles, Diffusion of general data on non-flat manifolds via harmonic maps theory: the direction diffusion case, *Internat. J. Comput. Vision* 36 (2) (2000) 149–161.
- [17] R. Tsai, L.-T. Cheng, S.J. Osher, H.K. Zhao, Fast sweeping method for a class of Hamilton–Jacobi equations, *SIAM J. Numer. Anal.* 41 (2003) 673–694.
- [18] C.L. Wu, J.S. Deng, W.M. Zhu, F.L. Chen, Inpainting images on implicit surfaces, *Pacific Graphics 2005, Macau, 2005*, pp. 142–144.