Multistencils Fast Marching Methods: A Highly Accurate Solution to the Eikonal Equation on Cartesian Domains

M. Sabry Hassouna, *Member*, *IEEE*, and Aly A. Farag, *Senior Member*, *IEEE*

Abstract—A wide range of computer vision applications require an accurate solution of a particular Hamilton-Jacobi (HJ) equation known as the *Eikonal* equation. In this paper, we propose an improved version of the fast marching method (FMM) that is highly accurate for both 2D and 3D Cartesian domains. The new method is called *multistencils fast marching* (MSFM), which computes the solution at each grid point by solving the Eikonal equation along several stencils and then picks the solution that satisfies the upwind condition. The stencils are centered at each grid point and cover its entire nearest neighbors. In 2D space, two stencils cover 8-neighbors of the point, whereas in 3D space, six stencils cover its 26-neighbors. For those stencils that are not aligned with the natural coordinate system, the Eikonal equation is derived using directional derivatives and then solved using higher order finite difference schemes. The accuracy of the proposed method over the state-of-the-art FMM-based techniques has been demonstrated through comprehensive numerical experiments.

Index Terms—Multistencils fast marching methods, monotonically advancing fronts, fast marching methods, level set methods, Eikonal equation.

1 Introduction

A large number of applications require the development of optimal algorithms for tracking moving interfaces (that is, advancing fronts). Advances in numerical analysis have led to computationally efficient tools for tracking interface motion by using level set methods [1]. Level set methods are numerical algorithms for tracking moving interfaces, whose motion is governed by a particular class of first-order hyperbolic partial differential equation (PDE) known as the Hamilton-Jacobi (HJ) equation. Level set methods possess several tempting features, which make them very appealing in several research areas. That they are applicable in higher dimensions, automatically handle topological changes of moving interfaces such as break and merge, and use efficient numerical schemes are among those features.

One important and interesting HJ equation is the Eikonal equation,[1] which is a first-order nonlinear PDE whose solution tracks the motion of a monotonically advancing front. The solution of this particular equation is found to be very useful in a wide range of computer vision applications such as computing distance fields from one or more points [2], shaping from shading [3], shape offsetting [4], optimal path planning [3], [5], segmentation [6], registration [7], [8], and shape representation [9], [10].

Several methods have been proposed to solve the Eikonal equation [2], [11], [12], [13], [14], [15], [16], [17]. The most stable and consistent methods among those techniques are the fast sweeping method (FSM) and the fast marching method (FMM), which are applicable to both Cartesian domains [2], [17], [18] and triangulated surfaces [19], [20], [21].

The FSM [17], [21], [22], [23], [24], [25] is an iterative algorithm with optimal complexity that finds the numerical solution by using a nonlinear upwind method and Gauss-Seidel type iterations with alternating sweepings in predetermined directions. The FMM [2], [18], [19], [20] combines entropy-satisfying upwind schemes and a fast sorting technique to find the solution in one-pass algorithm.

A recent computational study [26] of the FMM and the FSM for solving the Eikonal equation has shown that, on realistic grids, the FSM is faster than the FMM for problems with simple geometry. However, on a fixed grid, for nonuniform problems and/or complex geometry, the situation may be reversed.

The main advantage of the FMM over existing techniques is that it explicitly maintains a narrow band that separates the grid points of known solutions from the unknown ones. Therefore, the solution at every grid point is computed in an order that is consistent with the way wave fronts propagate. This order information is very important in several applications such as ordered thinning skeletonization [10], [27], robotic path planning [5], and image inpainting [28]. Unfortunately, the FMM is not accurate along diagonal directions, and its computational complexity is not optimal.

[1] The Eikonal equation was derived about 150 years ago by Sir William Rowan Hamilton. The word *Eikonal* was introduced in 1895 by H. Burns. It comes from the Greek word "εικων," from which the modern word *icon* is derived. The equation's title is descriptive because it controls the formation of images in optical systems.

- *The authors are with the Computer Vision and Image Processing Laboratory (CVIP), Department of Electrical and Computer Engineering, Room 412, University of Louisville, Louisville, KY 40292. E-mail: msabry@cvip.uofl.edu, aly.farag@louisville.edu.*

Manuscript received 9 Apr. 2006; revised 5 Sept. 2006; accepted 16 Oct. 2006; published online 17 Jan. 2007.
Recommended for acceptance by G. Sapiro.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0278-0406.
Digital Object Identifier no. 10.1109/TPAMI.2007.1154.

Therefore, some approaches have been introduced to improve its accuracy [18], [29] and efficiency [30], [31].

In this paper, we propose a highly accurate method for solving the Eikonal equation on Cartesian domains. The proposed method is called *multistencils fast marching* (MSFM), which builds over the FMM. It computes the solution at each grid point by solving the Eikonal equation along several stencils that cover its entire neighbor points and then picks the solution that satisfies the upwind condition. For those stencils that are not aligned with the natural coordinate system, the Eikonal equation is derived using directional derivatives and then solved using higher order finite difference schemes. In [32], we have presented some preliminary results of this work.

This paper is organized as follows: In Section 2, the equation of the motion of monotonically advancing fronts is derived. In Section 3, the FMM is explained in detail. Section 4 reviews related methods that improve the FMM and Section 5 introduces the proposed MSFM. Section 6 describes the numerical experiments that test the accuracy and convergence of the proposed methods in both 2D and 3D Cartesian domains. Finally, Section 7 concludes the presented work.

2 Monotonically Advancing Fronts

Consider a closed interface Γ (that is, boundary) that separates one region from another. Γ can be a curve in 2D or a surface in 3D. Assume that Γ moves only in one direction with speed F and hence has a monotonically increasing or decreasing propagation according to the sign of F. In order to track the position of the front, the arrival time $T(\boldsymbol{x})$ of Γ is computed as it crosses each point $\boldsymbol{x}$. The equation of the motion of this special front can be derived, as described in [2], as follows: Consider the one-dimensional case, where $distance = speed \times time$, which can be expressed as

$$F = \frac{dx}{dT}. \tag{1}$$

In multiple dimensions, the equation of motion is given as

$$|\nabla T|F = 1, \quad T(\Gamma_0) = 0, \tag{2}$$

where the arrival time T of the initial position of the front is set to 0. If the speed depends only on the position $\boldsymbol{x}$, then the equation reduces to a nonlinear first-order PDE, which is known in *geometrical optics* as the Eikonal equation. Both Adalsteinsson and Sethian [2] and Tsitsiklis [33] have independently solved that equation in one-pass algorithm. The approach of Adalsteinsson and Sethian is based on an upwind scheme, whereas the approach of Tsitsiklis is based on an optimal control theory. The main advantage of [2] over [33] is that it is more general in higher dimensions.

3 The Fast Marching Method

Because the proposed MSFM builds over the FMM, this section explains in detail the FMM algorithm. According to (2), if the speed is only a function of the location, then the FMM is called isotropic. Otherwise, it is called anisotropic if the direction of propagation is also taken into account. This paper is concerned only with isotropic speed functions. For anisotropic FMM, see [34].

In 2D, the numerical approximation of $|\nabla T|$ that selects the physically correct vanishing viscosity weak solution is given by Godunov [35]

$$\begin{aligned}&\max(D_{ij}^{-x}T, -D_{ij}^{+x}T, 0)^2 + \\ &\max(D_{ij}^{-y}T, -D_{ij}^{+y}T, 0)^2 = \frac{1}{F_{ij}^2},\end{aligned} \tag{3}$$

where D_{ij}^{-} and D_{ij}^{+} are the standard backward and forward finite difference schemes, respectively, at location (i,j). If ∇T is approximated by a first-order finite difference scheme, then (3) can be rewritten as

$$\sum_{v=1}^{2} \max\left(\frac{T - T_v}{\Delta_v}, 0\right)^2 = \frac{1}{F^2}, \tag{4}$$

where $\Delta_1 = \Delta x$, $\Delta_2 = \Delta y$, $T = T_{i,j}$, $F = F_{ij}$, and

$$\begin{aligned}T_1 &= \min(T_{i-1,j}, T_{i+1,j})\\ T_2 &= \min(T_{i,j-1}, T_{i,j+1}).\end{aligned} \tag{5}$$

The solution of (4) is given as

- $T > \max(T_1, T_2)$. T is the maximum solution of the following quadratic equation

$$\sum_{v=1}^{2} \left(\frac{T - T_v}{\Delta_v}\right)^2 = \frac{1}{F^2}. \tag{6}$$

- $T_2 > T > T_1$. $T = T_1 + \frac{\Delta_1}{F}$.
- $T_1 > T > T_2$. $T = T_2 + \frac{\Delta_2}{F}$.

A systematic procedure for solving the quadratic discrete equation in any dimension is provided in [36]. The idea behind the FMM is to introduce an order in the selection of the grid points during computation of their solutions (arrival times) in a way similar to the Dijkstra shortest path algorithm [37]. This order is based on the causality relationship, which states that the arrival time T at any point depends only on the adjacent neighbors that have smaller values. During the evolution of the front, each grid point $\boldsymbol{x}$ is assigned one of the three possible tags:

1. *Known*. The computed travel time at $\boldsymbol{x}$ will not be changed later.
2. *Narrow band*. The computed travel time at $\boldsymbol{x}$ may be changed later.
3. *Far*. The travel time at $\boldsymbol{x}$ is not yet computed.

The FMM algorithm can be summarized as follows. Initially, all boundary points are tagged as *known*. Then, their nearest neighbors are tagged as *narrow band* after their arrival time is computed by solving (4):

1. LOOP. Among all *narrow-band* points, extract the point with minimum arrival time and change its tag to *known*.
2. Find its nearest neighbors that are either *far* or *narrow band*.
3. Update their arrival times by solving (4).
4. Go back to LOOP.

As a result of the update procedure (Step 3), either a *far* point is tagged as a *narrow band* or a *narrow-band* point gets assigned a new arrival time that is less than its previous value.

Although the FMM provides a stable and consistent solution to the Eikonal equation, it still has two limitations. First, the computational complexity of the method is high because it stores the solutions in a *narrow band* that is implemented using a sorted heap data structure. The complexity of maintaining the heap is $O(\log n)$, where n is the total number of grid points. Therefore, the total complexity of the method is $O(n \log n)$. Second, at each grid point x, the method employs a 4-point stencil to exploit only the information of the adjacent four neighbors to x, thus ignoring the information provided by diagonal points. As a consequence, the FMM suffers from a large numerical error along diagonal directions.

4 Improved Fast Marching Methods

Several methods have been proposed to improve the FMM for it to be either computationally efficient [30], [31] or more accurate [18], [29]. In the following sections, we review each of these methods.

4.1 The Higher Accuracy Fast Marching Method (HAFMM)

The HAFMM [18] improves the accuracy of the FMM by approximating the gradient by a second-order finite difference scheme whenever the arrival times of the neighbor points are available, but reverts to a first-order approximation in other cases. The second-order backward and forward finite difference approximations of the gradient along the x direction of a 2D lattice is given in (7) and (8), respectively,

$$D_{ij}^{-x} = \frac{3T_{i,j} - 4T_{i-1,j} + T_{i-2,j}}{2\Delta x}, \tag{7}$$

$$D_{ij}^{+x} = -\frac{3T_{i,j} - 4T_{i+1,j} + T_{i+2,j}}{2\Delta x}. \tag{8}$$

By substituting (7) and (8) into (3), we get

$$\sum_{v=1}^{2} \max\left(\frac{3}{2\Delta_v}(T - T_v), 0\right)^2 = \frac{1}{F^2}, \tag{9}$$

where

$$T_1 = \min\left(\frac{4T_{i-1,j} - T_{i-2,j}}{3}, \frac{4T_{i+1,j} - T_{i+2,j}}{3}\right), \tag{10}$$

$$T_2 = \min\left(\frac{4T_{i,j-1} - T_{i,j-2}}{3}, \frac{4T_{i,j+1} - T_{i,j+2}}{3}\right). \tag{11}$$

The method asserts two conditions to use a second-order approximation of the gradient; otherwise, it reverts to a first-order approximation.

1. The points that are two points away from (i, j) in each direction are *known*. For example, in the x direction, $T_{i-2,j}$ or $T_{i+2,j}$ must be *known*.
2. The points that are two points away from (i, j) in each direction must have less travel time than those points at one point away from (i, j). For example, in the x direction, $T_{i-2,j} \leq T_{i-1,j}$, and $T_{i+2,j} \leq T_{i+1,j}$.

4.2 The Shifted Grid Fast Marching (SGFM) Method

Danielsson and Lin [29] improved the accuracy of the FMM by introducing the *SGFM* technique. The main idea of the algorithm is to sample the cost function at half-grid positions; therefore, the cost is dependent on the marching direction. The update strategy for computing the arrival time at a neighbor point of a *known* point under the new scheme is derived from the optimal control theory in a fashion similar to Tsitsiklis' [33]. Therefore, the solution cannot make use of any higher order finite difference schemes. In 2D, they proposed two solution models. The first one uses 4-connected neighbors, which gives the same results as the FMM, whereas the second one uses 8-connected neighbors, which gives better results than FMM. In both models, the idea is the same, where the neighborhood around x is divided to either four quadrants or eight octants. The *known* points of each quadrant/octant compute the arrival time at x, which is then assigned the minimum value over all quadrant/octants. In 3D, no improvements over the FMM have been reported.

4.3 The Group Marching Method (GMM)

The GMM is a modified version of the FMM that advances a group of grid points simultaneously rather than sorting the solution in a *narrow band*. The GMM reduces the computational complexity of the FMM to $O(n)$ while maintaining the same accuracy. The method works as follows: A group of points G is extracted from the *narrow band* such that the points' travel times do not alter each other in the update procedure. The neighbor grid points of G join the *narrow band* after their travel times are computed. Finally, the points G are tagged as *known*.

4.4 Untidy Fast Marching Method (UFMM)

In [31], another method is proposed to improve the computational efficiency of the FMM by reducing its complexity to $O(n)$. The method implements the *narrow band* by a special data structure called *untidy priority queue* whose maintenance (insert/delete) operations cost $O(1)$ rather than a sorted heap, whose cost is $O(\log n)$. The queue is implemented using a dynamic circular array. Each computed arrival time value is quantized and used as an index to access the dynamic array. Each entry (bucket) of the circular array contains a list of points with similar T values. The quantization is used only to place the grid point in the queue, whereas the actual T value is used to solve the Eikonal equation when the grid point is selected. Therefore, errors can only occur due to a wrong selection order. The authors have shown that the error introduced due to the wrong selection order is of the same order of magnitude as the FMM.

5 Multistencils Fast Marching Methods

All related methods except [29] ignore the information provided by diagonal points and hence suffer from large numerical errors along diagonal directions. One can make use of the diagonal information of a point x in two ways. One is by using only one stencil that is centered at x and is always aligned with the natural coordinate system. The coordinate system is then rotated several times to intersect the lattice at diagonal grid points. The other is by using several stencils that are centered at x and covering its entire diagonal neighbor points. The gradient is then approximated using

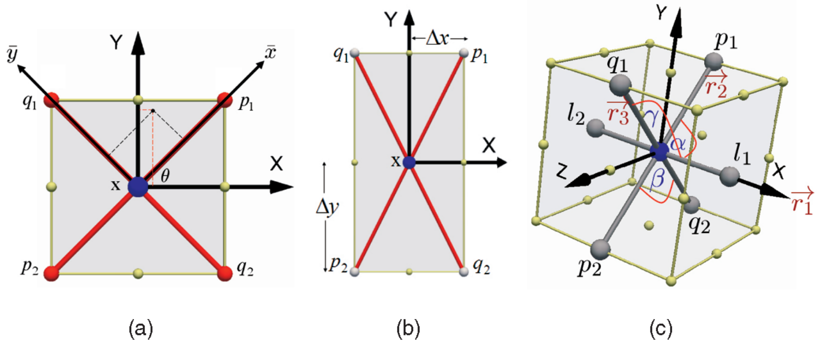


Fig. 1. (a) The natural coordinate system is rotated by an angle θ to intersect the lattice at diagonal grid points. (b) The stencil S_v is centered at x and intersects the lattice at diagonal grid points. (c) The stencil S_w is centered at x and intersects the lattice at the grid points l_1, l_2, p_1, p_2, q_1, and q_2.

directional derivatives. In the following, we will show that the latter method (that is, proposed work) is more general than the former. It is worth noting that using diagonal stencils to enhance the accuracy of solving PDEs is commonly used in practice [38], [39], [40]. The work presented in this paper combines multiple stencils and directional derivatives within the FMM to improve the accuracy of solving the Eikonal equation on Cartesian domains.

5.1 Single-Stencil Multiple Rotations of Coordinate System

Consider the 2D lattice in Fig. 1a. Assume that the Cartesian coordinate system has been rotated by an angle θ such that it intersects the lattice at grid points that were diagonal with respect to the original coordinate system p_1, p_2, q_1, and q_2. Any point $(\bar{x}, \bar{y})$ in the new coordinate system is related to a point (x, y) in the original one

$$\begin{aligned} \bar{x} &= \quad x\cos\theta + y\sin\theta \\ \bar{y} &= -x\sin\theta + y\cos\theta. \end{aligned} \tag{12}$$

The partial derivatives of T with respect to x and y are given by the chain rule

$$\begin{aligned} T_x &= \frac{\partial T}{\partial x} = \frac{\partial T}{\partial \bar{x}}\cdot\frac{\partial \bar{x}}{\partial x} + \frac{\partial T}{\partial \bar{y}}\cdot\frac{\partial \bar{y}}{\partial x} \\ T_y &= \frac{\partial T}{\partial y} = \frac{\partial T}{\partial \bar{x}}\cdot\frac{\partial \bar{x}}{\partial y} + \frac{\partial T}{\partial \bar{y}}\cdot\frac{\partial \bar{y}}{\partial y}. \end{aligned} \tag{13}$$

Substituting (12) into (13) results in

$$\begin{aligned} T_x &= \frac{\partial T}{\partial \bar{x}}\cos\theta - \frac{\partial T}{\partial \bar{y}}\sin\theta \\ T_y &= \frac{\partial T}{\partial \bar{x}}\sin\theta + \frac{\partial T}{\partial \bar{y}}\cos\theta. \end{aligned} \tag{14}$$

By squaring both sides of (14), it is easy to show that

$$T_x^2 + T_y^2 = T_{\bar{x}}^2 + T_{\bar{y}}^2 = |\nabla T(\bar{x}, \bar{y})|^2 = \frac{1}{F^2(x, y)}, \tag{15}$$

which is the Eikonal equation. Therefore, under the rotated coordinate system, ∇T can be approximated by the spatial derivatives along $\bar{x}$ and $\bar{y}$. Now, by solving the Eikonal equation once along the xy-coordinate system and once along the $\bar{x}\bar{y}$-coordinate system, two solutions for the arrival time at x are obtained, from which the smallest that satisfies the FMM causality relationship is selected. Unfortunately, this technique is limited only to an isotropic grid spacing, where $\Delta x = \Delta y$. Otherwise, the rotated coordinate system will intersect the lattice at nongrid points.

5.2 2D Multistencils Fast Marching Method

Consider the stencil S_v that intersects the 2D Cartesian domain at the grid points p_1, p_2, q_1, and q_2 as shown in Fig. 1b. Let $\vec{r_1} = [r_{11}\ r_{12}]^T$ and $\vec{r_2} = [r_{21}\ r_{22}]^T$ be the unit vectors along $\overline{p_2p_1}$ and $\overline{q_2q_1}$, respectively, and U_1 and U_2 be the directional derivatives along $\vec{r_1}$ and $\vec{r_2}$, respectively, which are given as

$$U_1 = \vec{r_1}\cdot\nabla T(\boldsymbol{x}) = r_{11}T_x + r_{12}T_y, \tag{16}$$

$$U_2 = \vec{r_2}\cdot\nabla T(\boldsymbol{x}) = r_{21}T_x + r_{22}T_y \tag{17}$$

and can be rewritten as

$$\begin{pmatrix} U_1 \\ U_2 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{pmatrix} \begin{pmatrix} T_x \\ T_y \end{pmatrix}. \tag{18}$$

Thus,

$$U = R\,\nabla T(\boldsymbol{x}), \tag{19}$$

$$\nabla T(\boldsymbol{x}) = R^{-1}U, \tag{20}$$

$$(\nabla T(\boldsymbol{x}))^T = (R^{-1}U)^T = U^T\,R^{-T}. \tag{21}$$

Since

$$|\nabla T(\boldsymbol{x})|^2 = (\nabla T(\boldsymbol{x}))^T\,\nabla T(\boldsymbol{x}), \tag{22}$$

then

$$|\nabla T(\boldsymbol{x})|^2 = U^T\,(RR^T)^{-1}U = \frac{1}{F^2(\boldsymbol{x})}. \tag{23}$$

If ϕ is the angle between the unit directional vectors, then $R\,R^T$ is given as

$$\begin{pmatrix} \|\vec{r_1}\|^2 & \vec{r_1}\cdot\vec{r_2} \\ \vec{r_2}\cdot\vec{r_1} & \|\vec{r_2}\|^2 \end{pmatrix} = \begin{pmatrix} 1 & \cos\phi \\ \cos\phi & 1 \end{pmatrix} \tag{24}$$

and

$$(R\,R^T)^{-1} = \frac{-1}{\sin^2\phi}\begin{pmatrix} -1 & \cos\phi \\ \cos\phi & -1 \end{pmatrix}. \tag{25}$$

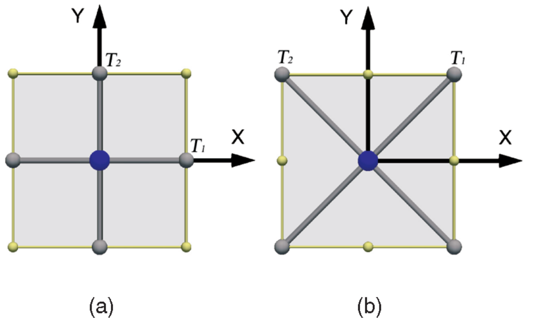


Fig. 2. The proposed stencils for the 2D Cartesian domain. T_1 and T_2 are the minimum arrival times along each arm of the stencil. (a) $\mathcal{S}_1$. (b) $\mathcal{S}_2$.

By substituting (25) into (23), a closed-form expression for the gradient of the arrival time along an arbitrary stencil is obtained as follows:

$$U_1^2 - 2U_1U_2\cos\phi + U_2^2 = \frac{\sin^2\phi}{F^2(\boldsymbol{x})}. \tag{26}$$

The first-order approximation of the directional derivative U_v that obeys the viscosity solution of (3) is given by

$$U_v = \max\left(\frac{T(\boldsymbol{x}) - T_v}{\|\boldsymbol{x} - \boldsymbol{x}_v\|}, 0\right), \quad v = 1, 2, \tag{27}$$

where $T(\boldsymbol{x})$, T_1, and T_2 are given in (5) and $\boldsymbol{x}_v$ is the *known* grid point at which T_v is minimum.

Also, the second-order approximation of the directional derivative U_v that obeys the viscosity solution of (3) is given by

$$U_v = \max\left(\frac{3}{2\,\|\boldsymbol{x} - \boldsymbol{x}_v\|}[T(\boldsymbol{x}) - T_v], 0\right), \quad v = 1, 2, \tag{28}$$

where T_1 and T_2 are given in (10) and (11), respectively, and $\boldsymbol{x}_v$ is still the *known* grid point at which T_v is minimum.

In 2D, two stencils $\mathcal{S}_1$ and $\mathcal{S}_2$ are used. The nearest neighbor points are covered by $\mathcal{S}_1$, whereas the diagonal points are covered by $\mathcal{S}_2$. To simplify the discussion, assume an isotropic grid spacing; $\Delta x = \Delta y = h$, then $\phi = 90$ degrees:

1. $\mathcal{S}_1$ Stencil: The stencil is aligned with the natural coordinate system as shown in Fig. 2a. Since $\phi = 90$, $(R\,R^T)^{-1} = I$ and, hence, (26) reduces to

$$U_1^2 + U_2^2 = \frac{1}{F^2(\boldsymbol{x})}. \tag{29}$$

For a first-order approximation of the directional derivative, the following equation is solved:

$$\sum_{v=1}^{2}\max\left(\frac{T(\boldsymbol{x}) - T_v}{h}, 0\right)^2 = \frac{1}{F^2(\boldsymbol{x})}. \tag{30}$$

On the other hand, for a second-order approximation of the directional derivative, the following equation is solved:

$$\sum_{v=1}^{2}\max\left(\frac{3}{2h}[T(\boldsymbol{x}) - T_v)], 0\right)^2 = \frac{1}{F^2(\boldsymbol{x})}. \tag{31}$$

TABLE 1
The Value of $g(h)$ as a Function of the Numerical Schemes and the Stencil Orientation

Stencil	1st-order scheme	2nd-order scheme
$\mathcal{S}_1$	$g(h) = 1/h^2$	$g(h) = 9/4h^2$
$\mathcal{S}_2$	$g(h) = 1/2h^2$	$g(h) = 9/8h^2$

2. $\mathcal{S}_2$ Stencil: The stencil is aligned with the diagonal neighbors as shown in Fig. 2b. Similar to the previous case, for a first-order approximation of the directional derivative, the following equation is solved:

$$\sum_{v=1}^{2}\max\left(\frac{T(\boldsymbol{x}) - T_v}{\sqrt{2}h}, 0\right)^2 = \frac{1}{F^2(\boldsymbol{x})}, \tag{32}$$

whereas, for a second-order approximation of the directional derivative, the following equation is solved:

$$\sum_{v=1}^{2}\max\left(\frac{3}{2\sqrt{2}h}[T(\boldsymbol{x}) - T_v], 0\right)^2 = \frac{1}{F^2(\boldsymbol{x})}. \tag{33}$$

For both stencils, if

$$T(\boldsymbol{x}) > \max(T_1, T_2), \tag{34}$$

then (30), (31), (32), and (33) reduce to a second-order equation of the form $\sum_{v=1}^{2} g(h)(T(\boldsymbol{x}) - T_v)^2$, which can be simplified to

$$g(h)\sum_{v=1}^{2} a_vT^2(\boldsymbol{x}) + b_vT(\boldsymbol{x}) + c_v = \frac{1}{F^2(\boldsymbol{x})}, \tag{35}$$

where the coefficients a_v, b_v, and c_v are given as

$$[\,a_v \quad b_v \quad c_v\,] = [\,1 \quad -2T_v \quad T_v^2\,]. \tag{36}$$

The value of $g(h)$ for the first and second-order numerical schemes, as well as the stencil orientation, is given in Table 1.

5.2.1 Upwind Condition

Solving the quadratic equation results in two solutions. The minimum is rejected because it does not maximize the equation. Still, the solution must satisfy the causality relationship. However, since the value of $T(\boldsymbol{x})$ is not known in advance, the computed solution $T(\boldsymbol{x})$ must be checked to see if it is higher than the two adjacent neighbors T_1 and T_2 that participated in the solution. If the check is true, then the solution is accepted. Otherwise, the following solution is accepted:

$$\min\left(T_v + \frac{\|\boldsymbol{x} - \boldsymbol{x}_v\|}{F(\boldsymbol{x})}\right), \quad v = 1, 2. \tag{37}$$

A better method is to derive the upwind condition before solving the equation. Since the solution is given as

$$T(\boldsymbol{x}) = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \tag{38}$$

TABLE 2
Coefficients of the Upwind Condition for both $\mathcal{S}_1$ and $\mathcal{S}_2$ Using Different Numerical Schemes

Stencil	1st-order scheme	2nd-order scheme
$\mathcal{S}_1$	$f = \min(\Delta x, \Delta y)$	$f = 2\min(\Delta x, \Delta y)$
$\mathcal{S}_2$	$f = \sqrt{\Delta^2 x + \Delta^2 y}$	$f = 2\sqrt{\Delta^2 x + \Delta^2 y}$

and the upwind condition requires that $T(\boldsymbol{x}) > T_1$ and $T(\boldsymbol{x}) > T_2$

$$T(\boldsymbol{x}) - T_v = \frac{-b + \sqrt{b^2 - 4ac}}{2a} - T_v > 0, \quad v = 1, 2. \tag{39}$$

After some algebraic manipulations, it is easy to show that the following condition must be satisfied:

$$aT_v^2 + bT_v + c < 0, \quad v = 1, 2. \tag{40}$$

The first and second-order approximations of the directional derivatives can be put in the form

$$U_v = k_v T(\boldsymbol{x}) - n_v, \quad v = 1, 2. \tag{41}$$

The values of k_v and n_v depend on the order of approximation and stencil orientation. Substituting (41) into (26) and solving for $T(\boldsymbol{x})$ result in the coefficient values of (40).

$$\begin{aligned} a &= k_1^2 - 2k_1k_2 \cos\phi + k_2^2 \\ b &= -2k_1n_1 + 2(k_1n_2 + k_2n_1)\cos\phi - 2k_2n_2 \\ c &= n_1^2 - 2n_1n_2\cos\phi + n_2^2 - \frac{\sin\phi^2}{F^2(\boldsymbol{x})}. \end{aligned} \tag{42}$$

Substituting (42) into (40) results in the following upwind condition:

$$|T_1 - T_2| < \frac{f(\Delta x, \Delta y)\sin\phi}{F(\boldsymbol{x})}. \tag{43}$$

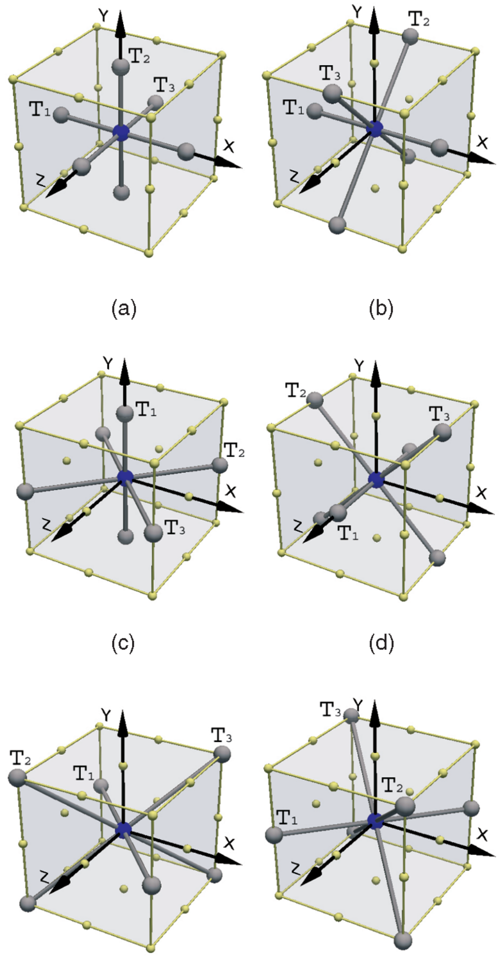
The value of $f(\Delta x, \Delta y)$ for the first and second-order numerical schemes, as well as the stencil orientation, is given in Table 2. In [25], [26], a more restrictive condition (44) is derived, which not only ensures the upwind condition, but also forces the quadratic solution to be real:

$$|T_1 - T_2| < \frac{f(\Delta x, \Delta y)(1 - \cos\phi)}{F(\boldsymbol{x})}. \tag{44}$$

5.3 3D Multistencils Fast Marching Method

The proposed 2D MSFM technique can be extended easily to a 3D Cartesian domain. Consider the stencil S_w that intersects the lattice at the grid points l_1, l_2, p_1, p_2, q_1, and q_2 as shown in Fig. 1c. Let $\overrightarrow{r_1} = [r_{11}\ r_{12}\ r_{13}]^T$, $\overrightarrow{r_2} = [r_{21}\ r_{22}\ r_{23}]^T$, and $\overrightarrow{r_3} = [r_{31}\ r_{32}\ r_{33}]^T$ be the unit vectors along $\overline{l_2 l_1}$, $\overline{p_2 p_1}$, and $\overline{q_2 q_1}$, respectively. Let α be the angle between $\overrightarrow{r_1}$ and $\overrightarrow{r_2}$, β be the angle between $\overrightarrow{r_2}$ and $\overrightarrow{r_3}$, and γ be the angle between $\overrightarrow{r_1}$ and $\overrightarrow{r_3}$. Finally, let U_1, U_2, and U_3 be the directional derivatives along $\overrightarrow{r_1}$, $\overrightarrow{r_2}$, and $\overrightarrow{r_3}$, respectively. Recall from Section 5.2 that

$$U^T\ (RR^T)^{-1}U = \frac{1}{F^2(\boldsymbol{x})}. \tag{45}$$



(a) (b) (c) (d) (e) (f)

Fig. 3. The proposed stencils for the 3D Cartesian domain. T_1, T_2, and T_3 are the minimum arrival times along each arm of the stencil. (a) $\mathcal{S}_1$. (b) $\mathcal{S}_2$. (c) $\mathcal{S}_3$. (d) $\mathcal{S}_4$. (e) $\mathcal{S}_5$. (f) $\mathcal{S}_6$.

Then,

$$R\,R^T = \begin{pmatrix} 1 & \cos\alpha & \cos\gamma \\ \cos\alpha & 1 & \cos\beta \\ \cos\gamma & \cos\beta & 1 \end{pmatrix}. \tag{46}$$

In 3D, six stencils $\mathcal{S}_w, w \in [1, 6]$ are used. The nearest neighbor points to $\boldsymbol{x}$ are covered by $\mathcal{S}_1$, whereas the diagonal points are covered by the rest of the stencils as shown in Fig. 3.

Again, to simplify the discussion, assume an isotropic grid spacing: $\Delta x = \Delta y = \Delta z = h$. Substituting $\alpha = \beta = \gamma = 90$ degrees into (46) results in $(R\,R^T)^{-1} = I$ and, hence,

$$U_1^2 + U_2^2 + U_3^2 = \frac{1}{F^2(\boldsymbol{x})}. \tag{47}$$

For a first-order approximation of the directional derivative, the following equation is solved:

$$\sum_{v=1}^{3} \max\left(\frac{T(\boldsymbol{x}) - T_v}{\|\boldsymbol{x} - \boldsymbol{x}_v\|}, 0\right)^2 = \frac{1}{F^2(\boldsymbol{x})}. \tag{48}$$

On the other hand, for a second-order approximation of the directional derivative, the following equation is solved:

TABLE 3
The Value of $g_v(h)$ as a Function of the Numerical Schemes and the Stencil Orientation

Stencil	1st-order scheme			2nd-order scheme		
	$g_1(h)$	$g_2(h)$	$g_3(h)$	$g_1(h)$	$g_2(h)$	$g_3(h)$
$\mathcal{S}_1$	$1/h^2$	$1/h^2$	$1/h^2$	$9/4h^2$	$9/4h^2$	$9/4h^2$
$\mathcal{S}_2$	$1/h^2$	$1/2h^2$	$1/2h^2$	$9/4h^2$	$9/8h^2$	$9/8h^2$
$\mathcal{S}_3$	$1/h^2$	$1/2h^2$	$1/2h^2$	$9/4h^2$	$9/8h^2$	$9/8h^2$
$\mathcal{S}_4$	$1/h^2$	$1/2h^2$	$1/2h^2$	$9/4h^2$	$9/8h^2$	$9/8h^2$
$\mathcal{S}_5$	$1/2h^2$	$1/3h^2$	$1/3h^2$	$9/8h^2$	$9/12h^2$	$9/12h^2$
$\mathcal{S}_6$	$1/2h^2$	$1/3h^2$	$1/3h^2$	$9/8h^2$	$9/12h^2$	$9/12h^2$

$$\sum_{v=1}^{3} \max\left(\frac{3}{2\|\boldsymbol{x}-\boldsymbol{x}_v\|}[T(\boldsymbol{x})-T_v],0\right)^2 = \frac{1}{F^2(\boldsymbol{x})}. \tag{49}$$

For all stencils, if

$$T(\boldsymbol{x}) > \max(T_1, T_2, T_3), \tag{50}$$

then (48) and (49) reduce to a second-order equation of the form $\sum_{v=1}^{3} g_v(h)(T(\boldsymbol{x})-T_v)^2$, which can be simplified to

$$\sum_{v=1}^{3} g_v(h)\big(a_v T^2(\boldsymbol{x}) + b_v T(\boldsymbol{x}) + c_v\big) = \frac{1}{F^2(\boldsymbol{x})}, \tag{51}$$

where the coefficients a_v, b_v, and c_v are given in (36). The value of $g_v(h)$ for the first and second-order numerical schemes, as well as the stencil orientation, is given in Table 3.

5.3.1 Upwind Condition

Although deriving the upwind condition for a 3D domain is straightforward, it has a complicated closed-form solution and hence is time consuming. A less expensive method is to make sure that the computed $T(\boldsymbol{x})$ is higher than the values of the three adjacent neighbors T_1, T_2, and T_3 that participated in the solution. If the check is true, then the solution is accepted. Otherwise, $T(\boldsymbol{x})$ is checked if it is greater than the values of the remaining two adjacent points. If the check is true, then the quadratic equation is solved based on their values, and the maximum solution is returned. Otherwise, the following solution is accepted:

$$\min\left(T_v + \frac{\|\boldsymbol{x}-\boldsymbol{x}_v\|}{F(\boldsymbol{x})}\right), \quad v = 1, 2, 3. \tag{52}$$

6 Numerical Experiments

In order to study the accuracy and convergence of the proposed methods in both 2D and 3D Cartesian domains, several numerical experiments have been conducted.

6.1 Accuracy

The following experiments compare the accuracy of the proposed first-order MSFM (MSFM_1) and the second-order MSFM (MSFM_2) with the first-order FMM (FMM_1) (that is, FMM), the second-order FMM (FMM_2) (that is, HAFMM), and the SGFM.

The exact analytical solution of the Eikonal equation, given a particular speed model, is assumed to be the gold standard. Since the analytical solution is hard to find at least for complex speed models, we started from a continuous and differentiable function $T_a(\boldsymbol{x})$ as described in [29] and then compared the computed arrival time $T(\boldsymbol{x})$ by each method, given a particular speed function with $T_a(\boldsymbol{x})$, by using different error norms. The speed is derived from the following equation:

$$F = \frac{1}{|\nabla T_a|}. \tag{53}$$

TABLE 4
The Ground Truth Arrival Times $T_a(\boldsymbol{x})$ for both 2D and 3D Cartesian Domains

$T_a(\boldsymbol{x})$	Closed Form Solution
$T_1(\boldsymbol{x})$	$\sqrt{(x-x_0)^2+(y-y_0)^2}-1$
$T_2(\boldsymbol{x})$	$\frac{(x-x_0)^2}{100}+\frac{(y-y_0)^2}{20}$
$T_3(\boldsymbol{x})$	$\frac{x^2}{10}\exp(\frac{y}{20})+\frac{y^2}{50}$
$T_4(\boldsymbol{x})$	$1-\cos(\frac{x-x_0}{20})\cos(\frac{y-y_0}{20})$
$T_5(\boldsymbol{x})$	$\sqrt{(x-x_0)^2+(y-y_0)^2+(z-z_0)^2}-1$
$T_6(\boldsymbol{x})$	$\frac{(x-x_0)^2}{100}+\frac{(y-y_0)^2}{20}+\frac{(z-z_0)^2}{20}$

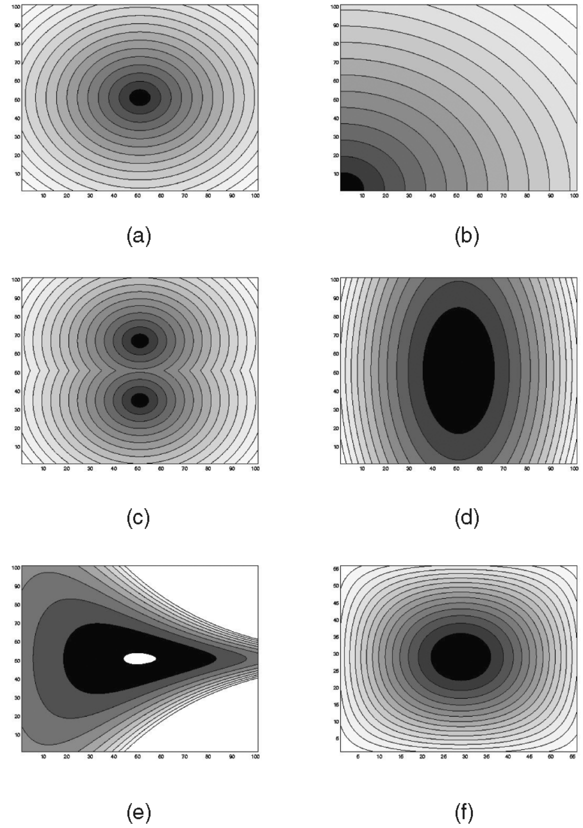


Fig. 4. Isocontours of $T_a(x)$. (a), (b), and (c) T_1 is generated from different source points. (d) T_2. (e) T_3. (f) T_4.

TABLE 5
Error Norms of the Computed T_1 from Different Source Points of a Grid of Size 101×101

Time $T(x)$	**T_1**								
Source Point(s)	**(51,51)**			**(1,1)**			**(51,35) and (51,67)**		
Method/Error	**L_1**	**L_2**	**L_∞**	**L_1**	**L_2**	**L_∞**	**L_1**	**L_2**	**L_∞**
FMM$_1$	0.746104	0.697090	1.314846	0.859562	0.951622	1.541677	0.617013	0.496484	1.200993
MSFM$_1$	0.291622	0.112154	0.577841	0.350694	0.167881	0.719245	0.274036	0.098130	0.577841
FMM$_2$	0.352778	0.148263	0.580275	0.346720	0.148236	0.586300	0.300247	0.110171	0.536088
MSFM$_2$	**0.069275**	**0.007711**	**0.215501**	**0.058817**	**0.004436**	**0.212368**	**0.074058**	**0.009852**	**0.420539**

Table 4 lists the ground truth arrival times for both 2D and 3D Cartesian domains. The isocontours of the ground truth functions for 2D spaces are shown in Fig. 4.

6.1.1 Experiment 1

This experiment compares the accuracy of the proposed methods MSFM$_1$ and MSFM$_2$ against FMM$_1$ and FMM$_2$ in 2D space under three different speed functions. The size of the test grid is 101×101 points, with $\Delta x = \Delta y = 1$. The first speed function is given as $F_1(\boldsymbol{x})$, which corresponds to a moving front from the point (x_0, y_0) with a unit speed. In this case, the computed arrival time $T_1(\boldsymbol{x})$ corresponds to the euclidean distance field, which is of interest in many scientific areas. The test is performed three times from one or more source points: one point at the center of the grid to test the high-curvature solution, one point at the corner of the grid to test the smooth solution, and two points to test the solution with shock points. To measure the error between the computed $T(\boldsymbol{x})$ and the analytical $T_a(\boldsymbol{x})$ solutions, the following error norms were employed, where n is the total number of grid points:

$$L_1 = \frac{1}{n}\sum_{i=1}^{n} |T - T_a|, \tag{54}$$

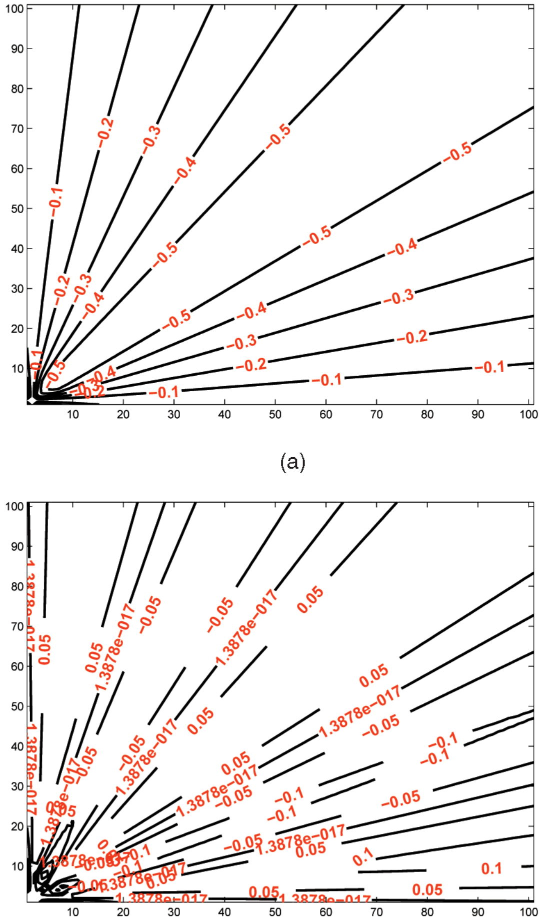
$$L_2 = \frac{1}{n}\sum_{i=1}^{n} |T - T_a|^2, \tag{55}$$

$$L_\infty = \max(|T - T_a|). \tag{56}$$

The numerical errors of these experiments are listed in Table 5. Fig. 5 shows the isocontours of the error curves when solving for T_1 in the first experiment by using the FMM$_2$ and the MSFM$_2$. Notice that the errors are small along the horizontal and vertical directions and increase rapidly in the regions around 45 degrees when using the FMM$_2$ as shown in Fig. 5a. However, when using the proposed method, the errors are small along all the directions that are covered by the proposed stencils as shown in Fig. 5b. Fig. 6 magnifies a portion of the isocontours of the exact solution (solid), the FMM$_2$ (dashed dot), and the MSFM$_2$ (dashed) for two different waves. The first wave propagates from the center of a grid (51, 51), whereas the second one propagates from two different source points (51, 35) and (51, 67). It is obvious that, in both cases, the proposed MSFM$_2$ is more accurate than the FMM$_2$. Also, the computed isocontours by MSFM$_2$ is nearly aligned with the exact analytical solution. Table 6 lists the numerical errors in the solution by two different complex speed models (F_2 and F_3) that are functions of the spatial coordinates.

6.1.2 Experiment 2

In [29], the authors have tested their method in 2D space by using different test functions (F_1 and F_4) on an isotropic grid of size 56×56. This experiment presents a comparison of the proposed methods MSFM$_1$ and MSFM$_2$, with their approach under the same experimental conditions. The numerical errors are listed in Table 7.



(a)

(b)

Fig. 5. Isocontour of the error curves of a unit speed wave that propagates from the corner of a grid when applying (a) the FMM$_2$ and (b) the proposed MSFM$_2$.

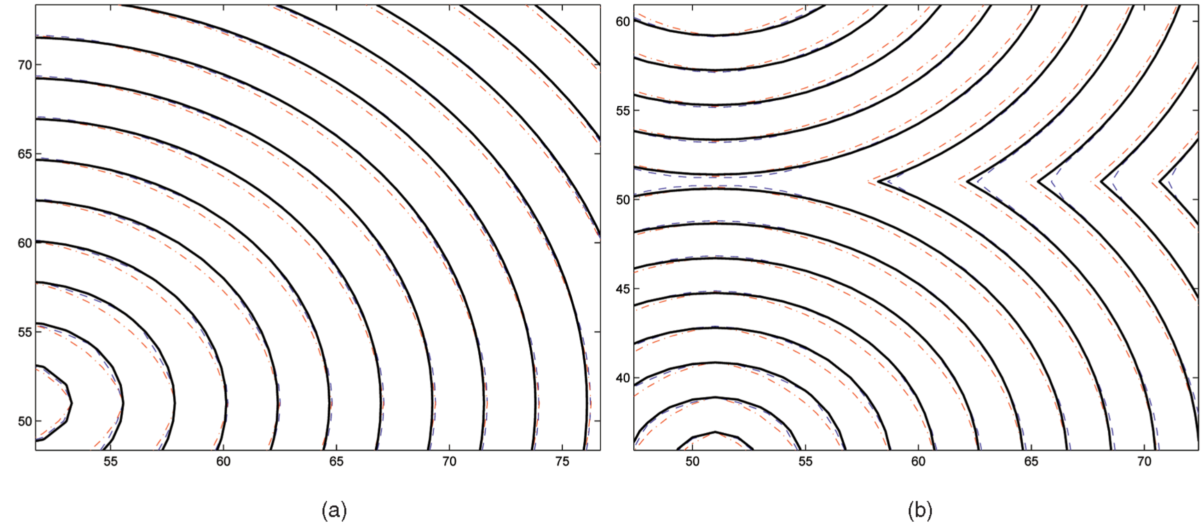


Fig. 6. Isocontours of the exact solution (solid), the FMM_2 (dashed dot), and the proposed MSFM_2 (dashed) for a unit speed wave that propagates from (a) one source point (51, 51) and (b) two source points at (51, 35) and (51, 67).

6.1.3 *Experiment 3*

This experiment tests the accuracy in 3D space under different speed functions (F_5 and F_6). The size of the test grid is $41 \times 41 \times 41$, with $\Delta x = \Delta y = \Delta z = 1$. No 3D results have been reported in [29]. The first speed function is given as $F_5(\boldsymbol{x})$, which corresponds to a moving front from the point (x_0, y_0, z_0) with a unit speed. Again, the test was performed three times from different points to test the quality of a high-curvature solution, a smooth solution, and a solution with shock points. The numerical errors for the first two cases are listed in Table 8.

In Table 9, we show the error norms of the computed arrival times in both 2D and 3D of a unit speed wave that propagates from the center of anisotropic grids. In 2D, $\Delta x = 0.1$ and $\Delta y = 0.2$, which correspond to a set of elliptical isocontours arrival times rather than circular ones. In 3D, $\Delta x = 0.1$, $\Delta y = 0.1$, and $\Delta z = 0.2$, which correspond to a set of ellipsoidal isosurfaces arrival times rather than spherical ones. To conclude, the numerical error tables for both isotropic and anisotropic grids in both 2D and 3D show that the proposed methods give the most accurate results among all related techniques, especially when using the MSFM_2. Notice also that the MSFM_1 is more accurate than the original FMM_1.

Fig. 7 shows a visual comparison of the computed 3D arrival times by the proposed MSFM_2 and the related methods FMM_1 and FMM_2 on a coarse grid of size $41 \times 41 \times 41$. It is clear that the proposed MSFM_2 provides a better high-curvature solution than the related methods.

Fig. 8 shows the computed 3D arrival times by the proposed method MSFM_2 for different speed functions from one or more source points. In Fig. 8a, the wave propagates with F_1 from the corner of the grid. In Fig. 8b, the wave propagates with F_1 from two different source points in the grid.

6.2 Convergence

A numerical method is said to be convergent if the numerical solution approaches the exact solution as the grid size h goes to 0. This section examines the convergence of the proposed methods MSFM_1 and MSFM_2 under different grid sizes in both 2D and 3D spaces. The test function represents a wave that propagates with a unit speed from the corner of each grid.

Theoretically speaking, as the grid size becomes finer in resolution, both MSFM_1 and MSFM_2 must converge to 1 and 2, respectively, because the former method approximates the gradient by using a first-order scheme, whereas the latter method uses a second-order scheme. Recall that the numerical error is proportional to h^p; then, p is the rate of convergence, which can be computed practically as

$$p = \frac{\log(L_i/L_{i+1})}{\log(h_i/h_{i+1})}, \tag{57}$$

TABLE 6
Error Norms of the Computed T_2 and T_3 from the Center of a Grid of Size 101×101

Time $T(\boldsymbol{x})$	**T_2**			**T_3**		
Source Point(s)	**(51,51)**			**(51,51)**		
Method/Error	**L_1**	**L_2**	**L_∞**	**L_1**	**L_2**	**L_∞**
FMM_1	1.514851	2.847438	3.000000	6.442591	120.800251	63.109117
MSFM_1	1.514851	2.847438	3.000000	6.297684	120.221017	63.109040
FMM_2	0.114048	0.013165	0.120108	0.388426	0.243170	1.758151
MSFM_2	**0.047245**	**0.002503**	**0.104651**	**0.160555**	**0.052614**	**1.356844**

TABLE 7
Error Norms of the Computed T_1 and T_4 from the Center of a Grid of Size 56×56

Time $T(\boldsymbol{x})$	**T_1**			**T_4**		
Source Point(s)	**(23,23)**			**(23,23)**		
Method/Error	**L_1**	**L_2**	**L_∞**	**L_1**	**L_2**	**L_∞**
SGFM	n/a	0.0276	0.2754	n/a	0.0504	0.5245
MSFM_1	0.235839	0.073034	0.468195	0.023620	0.000599	0.032788
MSFM_2	**0.063551**	**0.006169**	**0.205003**	**0.000737**	**0.000001**	**0.002494**

TABLE 8
Error Norms of the Computed T_5 and T_6 from the Center and the Corner of a Grid of Size $41 \times 41 \times 41$

Time $T(x)$	**T_5**						**T_6**		
Source Point(s)	**(21,21,21)**			**(1,1,1)**			**(1,1,1)**		
Method/Error	**L_1**	**L_2**	**L_∞**	**L_1**	**L_2**	**L_∞**	**L_1**	**L_2**	**L_∞**
FMM$_1$	2.062302	4.474762	3.527375	2.378944	6.027533	4.177612	2.226626	5.648937	4.719272
MSFM$_1$	0.751025	0.610959	1.224763	1.612016	2.923315	3.331898	1.918668	4.264334	3.943005
FMM$_2$	1.417481	2.154943	2.825411	0.833602	0.764195	1.360335	0.489881	0.329703	1.748241
MSFM$_2$	**0.266359**	**0.076616**	**0.432238**	**0.116653**	**0.017900**	**0.497072**	**0.054968**	**0.004366**	**0.214837**

where L_i is the error norm at grid size h_i. It is obvious from Tables 10 and 11 that the proposed methods in both 2D and 3D spaces converge to the optimal convergence rates as the grid size becomes finer.

6.3 Computational Complexity

The worst case complexity of the proposed method is still similar to that of the FMM, which is $O(n \log n)$. However, the computational time is higher. The computational complexity can be reduced to $O(n)$ by implementing the *narrow band* as an untidy priority queue [31].

In 2D space, the computational time is approximately 1-1.5 times higher than that of the FMM. However, in 3D space, the computational time is approximately three times higher than the FMM. In 3D, each grid point x has six neighbor points that share a face (*F-connected*), 12 neighbor points that share an edge (*E-connected*), and eight neighbor points that share a vertex (*V-connected*). According to the proposed stencils, $\mathcal{S}_1$ covers the *F-connected* points (6-neighbors), $\mathcal{S}_w$, $w \in [1, 4]$ cover both the *F-connected* and *E-connected* points (18-neighbors), and $\mathcal{S}_w$, $w \in [1, 6]$ cover the entire 26-connected neighbors. To strike a balance between high accuracy and less computational time ($\simeq$ two times higher than that of the FMM), the proposed methods in 3D can be restricted to use the first four stencils, which cover its 18-connected neighbor points.

TABLE 9
Error Norms of the Computed Arrival Time of a Unit Speed Wave that Propagates from the Center of 2D (101 $\times$ 101) and 3D $(41 \times 41 \times 41)$ Anisotropic Grids

Time $T(x)$	**T_1**			**T_5**		
Source Point(s)	**(51,51)**			**(21,21,21)**		
Method/Error	**L_1**	**L_2**	**L_∞**	**L_1**	**L_2**	**L_∞**
FMM$_1$	0.093769	0.106988	0.177274	0.693620	0.626133	1.610443
MSFM$_1$	0.046374	0.059139	0.133076	0.101452	0.012583	0.296870
FMM$_2$	0.043759	0.049431	0.083285	0.589116	0.451326	1.374615
MSFM$_2$	**0.017541**	**0.022933**	**0.052773**	**0.024059**	**0.001020**	**0.177943**

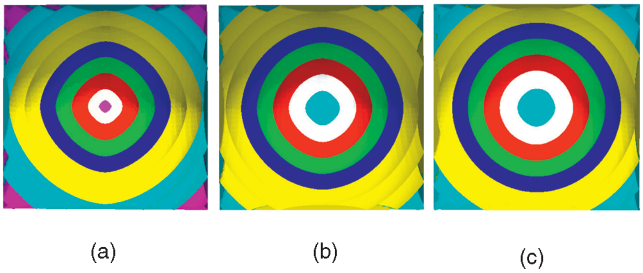


Fig. 7. Cross-sections in the arrival time field of a 3D unit speed wave that propagates from the center of a coarse grid of size $41 \times 41 \times 41$ by using (a) FMM_1, (b) FMM_2, and (c) the proposed MSFM_2. Notice that the proposed MSFM_2 provides a better high-curvature solution than the related methods.

7 Conclusion

This paper introduces the MSFM method as a highly accurate version of the original isotropic FMM for solving the Eikonal equation in both 2D and 3D Cartesian domains. At each grid point, the method solves the Eikonal equation along several stencils that cover the entire nearest neighbors of the point and then picks the solution that satisfies the upwind condition. For those stencils that are not aligned with the natural coordinate system, the Eikonal equation is derived using directional derivatives and then solved using higher order finite difference schemes. The computational complexities of the proposed methods are $O(n \log n)$; however, they can be reduced to $O(n)$ by implementing the *narrow band* as an untidy priority queue [31].

In the future, we intend to implement the proposed methods in the graphical processing unit (GPU) of the computer's graphics card to alleviate its computational time.

Acknowledgments

This research has been supported by the US National Science Foundation (NSF) Grant IIS-0513974. The authors would like to thank the anonymous reviewers for their comments and insights which improved the manuscript.

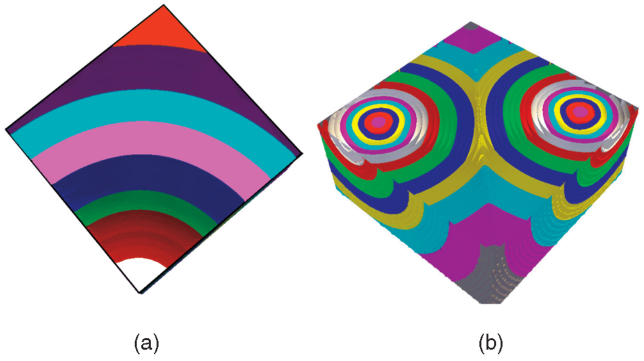


Fig. 8. Cross sections in the arrival time field of a 3D unit speed wave that propagates by the proposed method MSFM_2 from (a) a single source point and (b) two source points.

TABLE 10
Convergence of the Proposed Methods in 2D Space

	MSFM$_1$						**MSFM$_2$**					
h	L_1	p	L_2	p	L_∞	p	L_1	p	L_2	p	L_∞	p
1/21	3.85e-03	-	5.07e-03	-	1.07e-02	-	1.76e-03	-	2.18E-03	-	3.51e-03	-
1/41	2.70e-03	0.51	3.42e-03	0.56	7.22e-03	0.57	6.32e-04	1.48	7.28e-04	1.58	1.39e-03	1.33
1/81	1.57e-03	0.78	1.97e-03	0.79	4.21e-03	0.78	2.03e-04	1.63	2.34e-04	1.64	4.95e-04	1.5
1/161	8.47e-04	0.89	1.06e-03	0.89	2.30e-03	0.87	5.78e-05	1.81	6.67e-05	1.81	1.44e-04	1.78
1/321	4.40e-04	0.94	5.52e-04	0.95	1.21e-03	0.92	1.54e-05	1.90	1.78e-05	1.90	3.85e-05	1.90
1/641	2.24e-04	0.97	2.81e-04	0.97	6.30e-04	0.95	4.02e-06	1.95	4.62e-06	1.95	9.94e-06	1.96
1/1281	1.13e-04	0.99	1.42e-04	0.99	3.22e-04	0.97	1.02e-06	1.97	1.17e-06	1.97	2.52e-06	1.98
1/2561	5.56e-05	1.02	6.67e-05	1.09	1.61e-04	1.00	2.59e-07	1.98	2.98e-07	1.98	6.36e-07	1.99

T_1 is computed from the corner of the grid. The parameter p denotes the order of convergence.

TABLE 11
Convergence of the Proposed Methods in 3D Space

	MSFM$_1$						**MSFM$_2$**					
h	L_1	p	L_2	p	L_∞	p	L_1	p	L_2	p	L_∞	p
1/11	1.32e-02	-	1.53e-02	-	3.83e-02	-	5.96e-03	-	7.70e-03	-	1.71e-02	-
1/21	1.17e-02	0.18	1.35e-02	0.18	2.86e-02	0.42	3.12e-03	0.93	3.68e-03	1.1	7.14e-03	1.26
1/41	7.03e-03	0.73	7.88e-03	0.77	1.50e-02	0.93	1.10e-03	1.50	1.23e-03	1.58	2.29e-03	1.64
1/81	3.88e-03	0.86	4.31e-03	0.87	8.06e-03	0.89	3.30e-04	1.74	3.58e-04	1.78	6.33e-04	1.85
1/161	1.97e-03	0.98	2.20e-03	0.97	3.76e-03	1.1	9.10e-05	1.86	9.70e-05	1.88	1.72e-04	1.88

T_1 is computed from the corner of the grid. The parameter p denotes the order of convergence.

REFERENCES

[1] S. Osher and J. Sethian, "Fronts Propagating with Curvature Speed: Algorithms Based on Hamilton-Jacobi Formulations," *J. Computational Physics,* vol. 79, pp. 12-49, 1988.

[2] D. Adalsteinsson and J. Sethian, "A Fast Level Set Method for Propagating Interfaces," *J. Computational Physics,* vol. 118, pp. 269-277, 1995.

[3] R. Kimmel and J.A. Sethian, "Optimal Algorithm for Shape from Shading and Path Planning," *J. Math. Imaging and Vision,* vol. 14, no. 3, pp. 237-244, 2001.

[4] R. Kimmel and A.M. Bruckstein, "Shape Offsets via Level Sets," *Computer-Aided Design,* vol. 25, no. 5, pp. 154-162, 1993.

[5] M.S. Hassouna, A.E. Abdel-Hakim, and A. Farag, "Robust Robotic Path Planning Using Level Sets," *Proc. IEEE Int'l Conf. Image Processing,* pp. 473-476, 2005.

[6] T. Deschamps and L.D. Cohen, "Fast Extraction of Tubular and Tree 3D Surfaces with Front Propagation Methods," *Proc. Int'l Conf. Pattern Recognition,* pp. 731-734, 2002.

[7] Z. Cao, S. Pan, R. Li, R. Balachandran, M.J. Fitzpatrick, W.C. Chapman, and B.M. Dawant, "Registration of Medical Images Using an Interpolated Closest Point Transform: Method and Validation," *Proc. SPIE Int'l Soc. Optical Eng. Medical Imaging,* pp. 325-333, 2003.

[8] H.E.A.E. Munim and A.A. Farag, "A Shape-Based Segmentation Approach: An Improved Technique Using Level Sets," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 930-935, 2005.

[9] R. Kimmel, D. Shaked, N. Kiryati, and A. Bruckstein, "Skeletonization via Distance Maps and Level Sets," vol. 62, no. 3, pp. 382-391, 1995.

[10] M.S. Hassouna and A.A. Farag, "Robust Centerline Extraction Framework Using Level Sets," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 458-465, June 2005.

[11] V. Cerveny, "Ray Synthetic Seismograms for Complex Two- and Three-Dimensional Structures," *J. Geophysics,* vol. 58, pp. 2-26, 1985.

[12] J. Vidale, "Finite-Difference Calculation of Traveltimes in Three Dimensions," *J. Geophysics,* vol. 55, pp. 521-526, 1990.

[13] J. van Trier and W. Symes, "Upwind Finite-Difference Calculation of Traveltimes," *J. Geophysics,* vol. 56, pp. 812-821, 1991.

[14] P. Podvin and I. Lecomte, "Finite Difference Computation of Traveltimes in Very Contrasted Velocity Models: A Massively Parallel Approach and Its Associated Tools," *Geophysical J. Int'l,* vol. 105, pp. 271-284, 1991.

[15] S. Kim, "ENO-DNO-PS: A Stable, Second-Order Accuracy Eikonal Solver," *Soc. Exploration Geophysicists,* pp. 1747-1750, 1999.

[16] D.L. Chopp, "Some Improvements on the Fast Marching Method," *SIAM J. Scientic Computing,* vol. 23, no. 1, pp. 230-244, 2001.

[17] H. Zhao, "A Fast Sweeping Method for Eikonal Equations," *J. Math. and Computing,* vol. 74, no. 250, pp. 603-627, 2005.

[18] J. Sethian, *Level Sets Methods and Fast Marching Methods,* second ed. Cambridge Univ. Press, 1999.

[19] R. Kimmel and J. Sethian, "Fast Marching Methods on Triangulated Domains," *Proc. Nat'l Academy of Sciences,* vol. 95, no. 11, pp. 8341-8435, 1998.

[20] J.A. Sethian and A. Vladimirsky, "Fast Methods for the Eikonal and Related Hamilton-Jacobi Equations on Unstructured Meshes," *Proc. Nat'l Academy of Sciences,* vol. 97, no. 11, pp. 5699-5703, 2000.

[21] J. Qian, Y. Zhang, and H. Zhao, "Fast Sweeping Methods for Eikonal Equations on Triangulated Meshes," UCLA CAM Report 05-07, 2005, to appear in *SIAM J. Numerical Analysis.*

[22] R. Tsai, H. Zhao, and S. Osher, "Fast Sweeping Algorithms for a Class of Hamilton-Jacobi Equations," *SIAM J. Numerical Analysis,* vol. 41, no. 2, pp. 673-694, 2003.

[23] C.Y. Kao, S. Osher, and J. Qian, "Lax-Friedrichs Sweeping Scheme for Static Hamilton-Jacobi Equations," *J. Computational Physics,* vol. 196, no. 1, pp. 367-391, 2004.

[24] Y.-T. Zhang, H.-K. Zhao, and J. Qian, "High Order Fast Sweeping Methods for Static Hamilton-Jacobi Equations," *J. Scientific Computing,* to appear.

[25] C. Kao, S. Osher, and R. Tsai, "Fast Sweeping Methods for Hamilton-Jacobi Equations," *SIAM J. Numerical Analysis,* vol. 42, pp. 2612-2632, 2005.

[26] P.A. Gremaud and C.M. Kuster, "Computational Study of Fast Methods for the Eikonal Equation," *SIAM J. Scientific Computing,* vol. 27, no. 6, pp. 1803-1816, 2006.

[27] S. Bouix, K. Siddiqi, and A. Tannenbaum, "Flux Driven Fly Throughs," *Proc. IEEE Computer Vision and Pattern Recognition,* pp. 449-454, June 2003.

[28] A. Telea, "An Image Inpainting Technique Based on the Fast Marching Method," *J. Graphics Tools,* vol. 9, no. 1, pp. 23-34, 2004.

[29] P.E. Danielsson and Q. Lin, "A Modified Fast Marching Method," *Proc. Scandinavian Conf. Image Analysis,* pp. 1154-1161, 2003.

[30] S. Kim, "An o(n) Level Set Method for Eikonal Equations," *SIAM J. Scientific Computing,* vol. 22, no. 6, pp. 2178-2193, 2001.

[31] L. Yatziv, A. Bartesaghi, and G. Sapiro, "O(N) Implementation of the Fast Marching Algorithm," *J. Computational Physics,* vol. 212, pp. 393-399, Mar. 2006.

[32] M.S. Hassouna and A.A. Farag, "Accurate Tracking of Monotonically Advancing Fronts," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 355-362, June 17–22, 2006.

[33] J. Tsitsiklis, "Efficient Algorithms for Globally Optimal Trajectories," *IEEE Trans. Automatic Control,* vol. 40, no. 9, pp. 1528-1538, 1995.

[34] J.A. Sethian and A. Vladimirsky, "Ordered Upwind Methods for Static Hamilton-Jacobi Equations: Theory and Algorithms," *SIAM J. Numerical Analysis,* vol. 41, no. 1, pp. 325-363, 2003.

[35] S. Godunov, "Finite Difference Method for Numerical Computation of Discontinuous Solutions of the Equations of Fluid Dynamics," *Matematicheskii Sbornik,* pp. 47-271, 1959, translated from Russian by I. Bohachevsky.

[36] H. Zhao, "Fast Sweeping Method for Eikonal Equations," *Math. Computation,* vol. 74, pp. 603-627, 2004.

[37] E.W. Dijkstra, "A Note on Two Problems in Connection with Graphs," *Numerische Mathematik,* vol. 1, pp. 269-271, 1959.

[38] S. Chen, B. Merriman, S. Osher, and P. Smereka, "A Simple Level Set Method for Solving Stefan Problem," *J. Computational Physics,* vol. 138, no. 1, pp. 8-29, 1997.

[39] R. Kimmel and J. Sethian, "Computing Geodesic Paths on Manifolds," *Proc. Nat'l Academy of Sciences,* vol. 95, no. 15, pp. 8431-8435, 1998.

[40] J. Qian, Y. Zhang, and H. Zhao, "A Fast Sweeping Method for Static Convex Hamilton-Jacobi Equations," UCLA CAM Report 06-37, 2002, to appear in *J. Scientific Computing.*

M. Sabry Hassouna received the BSc and MSc degrees in electrical and computer engineering from Ain Shams University, Egypt, in 1995 and 1999, respectively. In 2006, he received the PhD degree in medical imaging from the Computer Vision and Image Processing Laboratory (CVIP Lab), University of Louisville, Kentucky. In the summer of 2006, he joined the research and development group of Vital Images Inc., Minnesota. He is a regular reviewer of a number of technical journals and conferences, including the *IEEE Transactions on Medical Imaging*, *IEEE Transactions on Image Processing*, and MICCAI. He is the inventor of fly-over, a new technology for virtual colonoscopy. His current research interests include computer vision, pattern recognition, and computer graphics with primary application to medical imaging, in which he has authored more than 20 technical articles. He was awarded the Outstanding ECE Graduate Student and the Dean's Citation Awards from the University of Louisville, the Excellence in Visualization Award from Silicon Graphics, and the Best Innovation Award on how to invest six billion on technology from Microsoft. He was the recipient of Research Louisville Award in medical imaging in 2003 and 2005. He is a member of the IEEE.

Aly A. Farag received the bachelor degree from Cairo University, Egypt and the PhD degree from Purdue University in electrical engineering. He joined the University of Louisville in August 1990, where he is currently a professor of electrical and computer engineering. At the University of Louisville, Dr. Farag founded the Computer Vision and Image Processing Laboratory (CVIP Lab) which focuses on imaging science, computer vision and biomedical imaging. Dr. Farag main research focus is 3D object reconstruction from multimodality imaging, and applications of statistical and variational methods for object segmentation and registration. He has authored or coauthored more than 250 technical papers in the field of image understanding, and coedited two volumes on *Deformable Models for Biomedical Applications* (Springer-Verlag 2007). He has applied his image understanding work to a number of industrial and biomedical applications including smart robotics, 3D reconstruction of the human jaw from video imaging, biomedical visualization, and in computer-assisted early detection of lung and colon cancer. He holds a number of patents in these applications. Dr. Farag was an associate editor of the *IEEE Transactions on Image Processing*. He is a regular reviewer for the US National Science Foundation and the US National Institutes of Health, and various technical journals and international conferences. He is a senior member of the IEEE and SME.