Applied Mathematics and Computation 218 (2011) 32–44

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

An adaptive domain-decomposition technique for parallelization of the fast marching method

Michael Breuß [a], Emiliano Cristiani [b,*], Pascal Gwosdek [a], Oliver Vogel [a]

[a] Faculty of Mathematics and Computer Science, Saarland University, Building E1.1, 66041 Saarbrücken, Germany
[b] Dipartimento di Matematica, SAPIENZA - Università di Roma, Rome, Italy

ARTICLE INFO

Keywords:
Parallel methods
Eikonal equation
Hamilton–Jacobi equations
Shape from Shading

ABSTRACT

The fast marching method (FMM) is an efficient technique to solve numerically the Eikonal equation. The parallelization of the FMM is not easy because of its intrinsic sequential nature. In this paper we propose a novel approach to parallelize the FMM. It leads to an equation-dependent domain decomposition and it turns out to be particularly suitable for machines with two or four cores that are in common use today. Compared to other techniques in the field, the proposed method is much simpler to implement and it gives a slightly better computational speed-up.

In order to test the new method on a real-world application, we solve the shape-from-shading problem based on a Hamilton–Jacobi equation. On a standard four-core machine, the method confirms the good properties. It shows a reasonable speedup factor of about 2.5, and it reveals its potential to good performance if the arithmetic density of the problem is high.

1. Introduction

The fast marching method (FMM) is an efficient technique to solve numerically the Eikonal equation

$$\begin{cases} f(x)|\nabla u(x)| = 1, & x \in \mathbb{R}^d \setminus \Gamma_0 \\ u(x) = u_0(x), & x \in \Gamma_0, \end{cases} \tag{1}$$

where $f(x) > 0$ is a given Lipschitz continuous function, and Γ_0 is a $(d-1)$-dimensional manifold in $\mathbb{R}^d$. Eq. (1) is well-posed in the framework of *viscosity solutions* [1]. The unique viscosity solution u of (1) is in general not differentiable, even if $f \in C^1(\mathbb{R}^d)$ and Γ_0 is smooth.

Eikonal and Eikonal-type equations appear in a number of different application fields [2], such as computer vision, image processing, optics, geoscience, and medical image analysis. In some cases, the approximation of the solution must be carried out on very large grids, requiring a significant computational time. For instance, this is the case of applications in computer vision such as shape from shading [3], or in image processing tasks such as inpainting [4] on real-size digital images. Although the FMM is much faster than a classic iterative algorithm where all the grid nodes are visited iteratively in a predefined order [5,6], solving Eq. (1) on large grids in real-time is still out of reach. In order to obtain a significant potential speed-up of the algorithm, an interesting option is to parallelize it. This paper proposes a novel parallel algorithm for FMM, which is both fast and easy to implement.

* Corresponding author.
E-mail addresses: breuss@mia.uni-saarland.de (M. Breuß), emiliano.cristiani@gmail.com (E. Cristiani), gwosdek@mia.uni-saarland.de (P. Gwosdek), vogel@mia.uni-saarland.de (O. Vogel).

0096-3003/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.amc.2011.05.041

Let us give a brief discussion of the FMM, restricting to the case of $d = 2$ to avoid cumbersome notations. FMM was introduced in [7–9], and it is based on Dijkstra's classic shortest path algorithm for graphs [10]. A complete proof of convergence can be found in [5], together with the right assumptions on the set-up that lets the method work in practice. First, a bounded computational domain $\Omega \supset \Gamma_0$ is introduced and discretized by a regular grid $G = \{(x_i, y_j); i = 1, \ldots, N_x; j = 1, \ldots, N_y\}$. Every cell is a square of side length Δx. Eq. (1) is then discretized by means of the usual upwind first-order finite-difference approximation introduced in [11]. u_{ij} denotes the approximation of the solution u at (x_i, y_j).

It can be shown that an iterative fixed-point algorithm based on this discretization converges, in a large number of iterations, to the viscosity solution [11]. The idea behind the FMM is to introduce an *ordering* in the selection of the grid nodes in such a way that convergence is reached in just one iteration over the grid.

The FMM realizes this as follows. During the computation, the grid G is always partitioned in three sets named *accepted*, *trial* and *far*, respectively. The *accepted* nodes are those where the solution has been already computed; an accepted value does not change any more. The *trial* nodes are the nodes where the computation actually takes place. Their value can still change as long as they are labelled as *trial*. Finally, the *far* nodes are the remaining nodes where an approximate solution has not been computed yet. For initialization, the nodes adjacent to Γ_0 are labelled as *accepted* and their value is set to u_0. All the values at non-*accepted* nodes adjacent to an *accepted* node are computed using the upwind scheme, and these nodes are labelled as *trial*. All the remaining nodes are labelled as *far* and their value is set to infinity, or just to a very large value. At every step of the algorithm, the *trial* node with the minimal value is labelled as *accepted*, and all its *far* neighbours are labelled as *trial*. Only the non-*accepted* neighbours of the last *accepted* node are computed. The accept-the-minimum rule is crucial, and it is based on the fact that a value can not be affected by other values larger than itself. The principle behind the latter property is called *causality principle*. The algorithm ends when all the nodes are *accepted*.

The FMM has been the subject of many researches since its introduction, leading to a number of improvements. Papers [12–14] proposed modifications to speed up the method and drop the computational complexity. Papers [5,15–17] deal instead with modifications leading to a higher accuracy of the approximate solution. FMM can be also adapted to solve more general Hamilton–Jacobi equations. This has been proven in [18], and then investigated in more detail in [19]. Meanwhile, several extensions of the FMM, such as in [18,20–24], were proposed.

Regarding parallelization techniques for the FMM the literature is quite scarce. This is due to the fact that the FMM works in a highly sequential way. Indeed, only one node per iteration becomes *accepted*, and nodes must be computed in a special order so that the causality principle is respected. Until now, [25–27] have been the only three notable works that discussed parallelization of the FMM. [25] proposes a parallel algorithm based on a classical domain-decomposition method. [26] attempted modifying the method in technical detail. [27] focused on a parallel implementation using graphics processing units (GPUs). However, while the algorithm presented in that paper is somewhat similar to the FMM, it is specifically tailored to geodesic distance computation. In this paper, the method as applied in [25] is considered, since the algorithmic basis and the assumed underlying computer architecture are identical to those of the proposed method.

There exist also other parallelization strategies for solving the Eikonal Eq. (1). One of these approaches is based on the Fast Sweeping Method [28] for which a parallelization was proposed in [29]. Another one is the Fast Iterative Method [30,31] which was parallelized on GPUs as described in the same papers.

This paper proposes a new parallelization technique for the FMM which creates automatically its own domain decomposition that depends on f, u_0, and Γ_0. The new method is much easier to implement than the one proposed in [25]. The idea adopted is to split the set Γ_0 (rather than Ω) among the processors since the very beginning of the computation. Then the subsets of Γ_0 resulting from this splitting are used by separate processors as starting points for parallel computations. The process interaction is realized by a relatively simple procedure constructed to satisfy the causality principle. Some numerical tests validate the usefulness of the new approach. It turns out to be particularly useful for machines with two to four cores and computationally competitive to the domain-decomposition technique employed in [25]. In order to test the new method on a real-world application, we solve a modern version of the shape-from-shading problem, which is a classic inverse problem in computer vision. The mathematical model adopted in this paper is based on a Hamilton–Jacobi equation which can be solved by means of FMM.

2. Domain-decomposition method

In this section, the domain-decomposition method (DDM) introduced in [25] is discussed briefly. Initially, the domain Ω is divided in P sub-domains $\mathcal{D}_1, \ldots, \mathcal{D}_P$, P being the number of processes to use in parallel. Each domain is extended by *ghost nodes* in the normal direction to the boundary, see Fig. 1. Ghost nodes are shared by neighbouring sub-domains and allow communication between corresponding processes.

The next step is to assign processes. After the usual initialization of the sequential FMM, each sub-domain is assigned to one process. Every process creates his own *trial* region, and starts working independently of the others. Basically, the load balance depends on the domain decomposition and the shape/position of Γ_0. It is possible that only one sub-domain contains Γ_0, and then, at the beginning of the computation, all processes but one are idle.

An important issue is communication between sub-domains. Every time a process updates a ghost node, the information is communicated to the sub-domain the node is shared with. In this way the information flowing along characteristics moves from a sub-domain to another, until the domain Ω is fully covered.