ARTICLE IN PRESS

Computers and Geosciences xxx (2017) 1–11

Contents lists available at ScienceDirect

Computers and Geosciences

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

Research paper

A Tracking Analyst for large 3D spatiotemporal data from multiple sources (case study: Tracking volcanic eruptions in the atmosphere)

Mohamed A. Gad[a,*], Mai H. Elshehaly[b], Denis Gračanin[b], Hicham G. Elmongui[b,c,d]

[a] Irrigation & Hydraulics Laboratory, Civil Engineering Department, Ain Shams University, Egypt
[b] Computer Science Department, Virginia Tech, USA
[c] Computer & Systems Engineering Department, Alexandria University, Egypt
[d] Deanship of Scientific Research, Umm Al-Qura University, Saudi Arabia

ARTICLE INFO

Keywords:
Tracking Analyst
Spatiotemporal
Interpolation
Data fusion
Volcanic eruptions

ABSTRACT

This research presents a novel Trajectory-based Tracking Analyst (TTA) that can track and link spatiotemporally variable data from multiple sources. The proposed technique uses trajectory information to determine the positions of time-enabled and spatially variable scatter data at any given time through a combination of along trajectory adjustment and spatial interpolation. The TTA is applied in this research to track large spatiotemporal data of volcanic eruptions (acquired using multi-sensors) in the unsteady flow field of the atmosphere. The TTA enables tracking injections into the atmospheric flow field, the reconstruction of the spatiotemporally variable data at any desired time, and the spatiotemporal join of attribute data from multiple sources. In addition, we were able to create a smooth animation of the volcanic ash plume at interactive rates. The initial results indicate that the TTA can be applied to a wide range of multiple-source data.

1. Introduction

Computational Fluid Dynamics (CFD) techniques such as Large-Eddy Simulation (LES) or Direct Numerical Simulations (DNS) can produce very large, time-varying, multi-field data sets. Exploration and analysis of these data sets are complicated processes due to their size, complexity and time-varying nature. Therefore, instead of saving the simulations into grid or finite element formats, numerical simulations of unsteady flow fields are usually stored as a set of point features organized in trajectories that pass through user-defined seed points (Lane, 1996; Max and Becker, 1999; McKenna et al., 2002; Konopka et al., 2007; and others). Particle tracing has been a central topic in flow visualization. The bulk of the work, however, has relied on a velocity field representation of the flow and has used numerical integration methods for the tracing process (Post et al., 2003; McLoughlin et al., 2010; and others). Since these integration-based techniques are computationally expensive and time consuming, techniques have been developed to efficiently sample the space and to use GPU parallelism to speed up the process (Schafhitzel et al., 2007; Burger et al., 2009). Kruger et al. (2005) advected particles on the GPU to allow for interactive visualization of steady flow on uniform grids to visualize streamlines and stream ribbons.

On the other hand, scatter observations of constituents transported in the flow field are usually made through direct measurements or remote sensing instruments. Examples of these constituents are suspended substances, pollutants, and water vapor in the atmosphere. It is always required to fuse the measurements from multiple sources to form a time continuum which becomes problematic when the measurements are not taken at the same times and locations. Such data fusion is important in developing meaningful visualizations, and can serve several purposes including spatiotemporal correction of orbital data and the resampling of data into structured formats (Kohrs et al., 2013), domain filling of missing data and plume tracking (Fairlie et al., 2014), and sensors cross correlations (Wu et al., 2017).

In cases where scatter observations from all involved sensors are referenced in the same time scale (i.e., regular in time) and completely cover the area of interest at every time step, the spatiotemporal problem is easier since it is reduced to spatial-only interpolation at every time step (Philip and Watson, 1982; Franke, 1982; Montmollin et al., 1980; Isaaks and Srivastava, 1989; and others). However, this is not the case in most of the large scale spatiotemporal interpolation domains, where multiple sensor types are involved. The problem of spatiotemporal data analysis and visualization then becomes much more complicated due to the fact that data is obtained from multiple sensor types, each with a different

* Corresponding author. Tel.: +2 01000 373088.
E-mail addresses: hydroshams@yahoo.ca (M.A. Gad), maya70@vt.edu (M.H. Elshehaly), gracanin@vt.edu (D. Gračanin), elmongui@alexu.edu.eg (H.G. Elmongui).

https://doi.org/10.1016/j.cageo.2017.10.003
Received 16 April 2017; Received in revised form 15 August 2017; Accepted 6 October 2017
Available online xxxx
0098-3004/

time scale. Further, sensor coverage is limited in both space and time. Data from polar orbiting satellites constitutes a clear example of such sensors since it has limited spatial coverage (strips) where the same region on Earth is visited once or twice a day depending on the satellite orbital speed. In such cases, spatiotemporal interpolation is needed to construct an instantaneous (i.e., at the same time) full scan of the whole Earth. The spatiotemporal problem becomes more complex when more than one satellite is involved. This paper develops an interpolation technique that makes use of trajectory information to perform the spatiotemporal interpolation in an attempt to provide a practical solution to fill this gap.

2. Problem statement

The TTA (Trajectory-based Tracking Analyst) proposes a novel spatiotemporal interpolation technique, called Trajectory-based Spatial and Temporal Interpolation (TSTI). The TSTI was first used by Elshehaly et al. (2014, 2015) for visualization purposes. The detailed description and application of the TSTI method is presented here in the current study. The method aims to interpolate motion field information at any given location not given in the original trajectories. The interpolated trajectories can then be used to move ("slide") the points of interest (whether they are remotely sensed detections or injected plume seeds in the flow field) to the corresponding positions at the desired target times. Unlike other established techniques (Vernier et al., 2013; Fairlie et al., 2014; and others), the analysis environment in the TTA is not the circulation model itself but the seeding is made completely outside the model (it only uses a trajectory data set obtained from the model without seeding the detections inside the model itself). This has the advantages of: *(i)* simplicity: the method works independently outside the model; *(ii)* efficiency: it runs quickly and can seed big data sets; *(iii)* flexibility: it requires only trajectory data whether from simulations or from RK4 integration of velocity vectors obtained from image cross correlation and pattern matching techniques; *(iv)* precision: the ability to seed high resolution scatter data, e.g., narrow plumes; and *(v)* practicality: the ability to seed and link different types of scatter data, i.e., different sensors, into the flow field which facilitates data joining operations.

3. Methodology

The developed TSTI technique has two components: (a) along trajectory adjustment (ADJUSTT) and (b) spatiotemporal interpolation (SPATIOT). The idea is to use motion information (i.e., the spatial translations) from the nearest m trajectories ($m = 4$–8) and spatially interpolate the translation information to the un-gauged location under consideration based on its relative location to the trajectories. A detailed description of the technique follows.

3.1. ADJUSTT

Data obtained from large unsteady state simulations is usually stored in the form of trajectories, or pathlines, each consisting of a stream of time-enabled points organized in a sequence of polylines. The attributes of the trajectory points include information about their position in 3D, a timestamp, and possibly a set of scalar values that are associated with each point from simulation results (e.g. temperature, pressure, etc.). Each polyline (i.e., trajectory) can be looked at as the locus of motion of a particle at the different times on a time scale. The time step along the trajectories ΔT is usually constant (or fractions of the constant). The objective of ADJUSTT is to determine the position along the same trajectory at any time instant within the time step. This is achieved through relative second degree polynomial fitting. To elaborate, consider a trajectory where point (X_{T1}, Y_{T1}, Z_{T1}) denotes a position at time $T1$ on a trajectory (Fig. 1), the position after $dt < \Delta T$ on the same trajectory (at time $T1+dt$) can be obtained by adding the spatial displacement (translation) vector (Equation (1)):

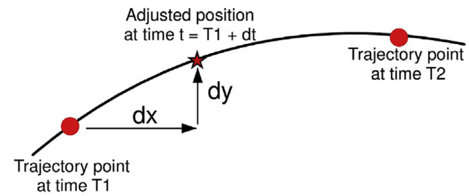

Fig. 1. ADJUSTT determines the position along trajectories at any required time t not coinciding on the time step ($\Delta T = T2\text{-}T1$) of the simulation data set (i.e., $dt < \Delta T$). This requires the pre-calculation of the coefficients of second degree polynomials (a_1, a_2, b_1, b_2, c_1, c_2). The second degree polynomials are used to predict the position along the trajectory in between the simulation time step. Note that the coefficients are pre-calculated at all simulation vertices using least squares and added as new attributes to simulation data sets. Three positions (the current position and the subsequent two positions) are used in the least squares to pre-determine the coefficients at all the points of the simulation.

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{T1+dt} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{T_1} + \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} \tag{1}$$

$$\begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} = \begin{pmatrix} a_1 & a_2 \\ b_1 & b_2 \\ c_1 & c_2 \end{pmatrix} \times \begin{pmatrix} dt \\ dt^2 \end{pmatrix} \tag{2}$$

where (dx, dy, dz) is the spatial displacement vector (obtained from Equation (2)) and a_1, a_2, b_1, b_2, c_1, c_2 are the coefficients of the second degree polynomials. These coefficients can be calculated using the least squares method, either on the fly during execution or during preprocessing, and added as attributes to the original data set. In this paper, we use the second option (pre-processing) to calculate the six coefficients at every point along all trajectories (i.e., determine the second degree polynomial that passes through every point and the next two points on all trajectories by solving a simple least squares matrix form at every trajectory point (Equation (3)):

$$\begin{pmatrix} 3 & \sum t & \sum t^2 \\ \sum t & \sum t^2 & \sum t^3 \\ \sum t^2 & \sum t^3 & \sum t^4 \end{pmatrix} \times \begin{pmatrix} 0 & 0 & 0 \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{pmatrix} = \begin{pmatrix} \sum dx & \sum dy & \sum dxz \\ \sum tdx & \sum tdy & \sum tdz \\ \sum t^2dx & \sum t^2dy & \sum t^2dz \end{pmatrix} \tag{3}$$

Since the first point is always the origin ($t = 0$, $dx = 0$, $dy = 0$, $dz = 0$) on the relative frame of reference, the least squares summations in Equation (3) are done for the next two points only where the second and third points are (t_1, dx_1, dy_1, dz_1) and (t_2, dx_2, dy_2, dz_2), respectively. Note that the spatial shifts are taken from the first point (i.e., the origin) while t_1 and t_2 are the differences in time with the first point and usually equal ΔT and $2\Delta T$ respectively (if the time step of the trajectory data set is constant).

3.2. SPATIOT

SPATIOT determines the translation during a certain period of any point of interest (e.g., sensor detection or a point not available in the simulation) by interpolating the corresponding shifts of the nearest trajectories points (in time and space) to the point of interest. The spatial interpolation is based on the inverse distance weighted (IDW) principle in which closer points are given much higher weights. In order to explain the interpolation, consider a point of interest at source time Ts for which we need to determine the corresponding position at destination time Td (note that this point of interest is not included in any of the simulation trajectories). On the other hand, the trajectory simulation dataset contains np polylines and total number of vertices N on a regular time scale