Computers and Geosciences 111 (2018) 283–293

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 19 October 2017

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)):

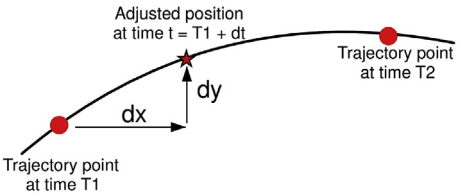


Fig. 1. ADJUSTT determines the position along trajectories at any required time t not coinciding on the time step (ΔT = $T2$-$T1$) of the simulation data set (i.e., $dt < \Delta T$). This requires the pre-calculation of the coefficients of second degree polynomials (a_1, a_2, b_1, b_2, c_1, c_2). The 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 pre-processing, 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

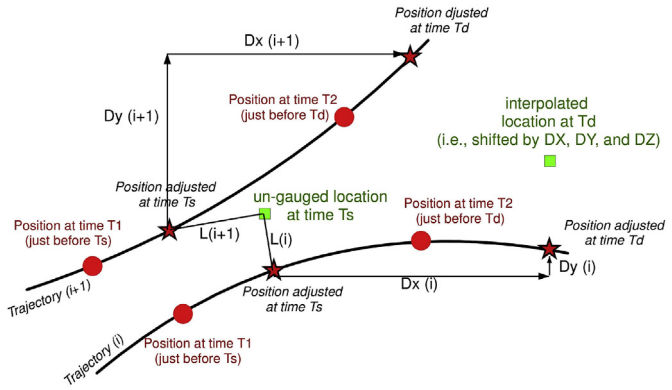


Fig. 2. SPATIOT finds the position of any detected point (not included in the simulation data set, i.e., un-gauged location) at any target times (T_d). It uses the inverse distance weighted spatial interpolation to determine the displacement of the detection from the displacements of the nearest neighborhood points (the nearest in time and space). n is the number of nearest neighborhood points from the simulation used in the interpolation (recommended n = 4–8, 8 is used). Note that ADJUST (Fig. 1) is used here to find the exact positions of the simulation points at both the time of the detection and the target time.

where *T1* and *T2* are the nearest times before *Ts* and *Td* respectively (Fig. 2). The interpolation proceeds as follows:

- Select all points at time *T1* from the simulation. Note that the number of selected points is equal to or less than the number of polylines *np* in the simulation data set. Then use ADJUSTT to slide the selected points from their positions at *T1* to their corresponding adjusted positions at *Ts* (source time) and select the nearest *m* positions from the *np* adjusted positions (recommended $m = 4$–8 points). Note that using small *m* values (less than 4) will not improve the accuracy by detecting abrupt and local details since the trajectories form a very smooth surface already with no abrupt changes. Actually decreasing m (less than 4) increases the chance of selecting the neighborhood from one side only of the detection which is not favorable. On the other side increasing *m* (more than 8) is also not desired to avoid unnecessary computations and unnecessary smoothing of the interpolated trajectories.
- Use ADJUSTT again to slide the *m* corresponding positions from time *T2* to time *Td* (i.e., slide the positions from time *T2* to their corresponding positions at *Td*).
- Find the *m* corresponding spatial shifts (Dx_i, Dy_i, and Dz_i) between the adjusted *m* positions at time *Ts* and the corresponding adjusted *m* positions at time *Td*.
- Use a spatial interpolation method to find the translation vector (*DX*, *DY*, *DZ*) between the point of interest and its corresponding position at *Td*. The IDW spatial interpolation method is used as follows (Equation (4)):

$$\begin{pmatrix} DX \\ DY \\ DZ \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{i=m} \frac{Dx_i}{L_i^2} \Big/ \sum_{i=1}^{i=m} \frac{1}{L_i^2} \\ \sum_{i=1}^{i=m} \frac{Dy_i}{L_i^2} \Big/ \sum_{i=1}^{i=m} \frac{1}{L_i^2} \\ \sum_{i=1}^{i=m} \frac{Dz_i}{L_i^2} \Big/ \sum_{i=1}^{i=m} \frac{1}{L_i^2} \end{pmatrix} \quad (4)$$

Note that (Dx_i, Dy_i, Dz_i) denote the spatial shifts of the nearest neighborhoods between their source and destination times (i varies from 1 to m); L_i denotes the distance between the nearest neighborhoods and the point under consideration (all at the same source time *Ts*); and (*DX*, *DY*, and *DZ*) denote the interpolated spatial shifts of the point under consideration between source and destination times.

4. Case study and data pre-processing

Volcanic eruptions eject ash and sulfur dioxide in the atmosphere. Ash threatens aviation while sulfur dioxide creates sulfate aerosol that affects the Earth's energy budget and climate.

On June 4, 2011 the Chilean Puyehue-Cordón Caulle Complex exploded injecting tons of ash in the southern atmosphere. This volcano belongs to the Andean mountains where the center of the main vent is located at 40.59°S and 72.117°W. The eruption arrived at the lower stratosphere, and ash was carried by the southern jet stream and circled the Earth. Accordingly, aviation in South America, Australia, and New Zealand stopped for safety. This eruption has been studied from different aspects in many studies (Castro et al., 2013; Schipper et al., 2013; Tuffen et al., 2013; Kluser et al., 2013; Collini et al., 2013; Bignami et al., 2014; and others).

The Michelson Interferometer for Passive Atmospheric Sounding (MIPAS) and the Atmospheric Infrared Sounder (AIRS) can provide direct measurements of the volcanic particles in the atmosphere. MIPAS can provide altitude information but with very poor horizontal resolution while AIRS provides good horizontal resolution with no altitude information. Data developed by both sensors suffers other problems due to the orbital time shift and data gaps. The datasets from the two satellites combined with an atmospheric simulation produced by the Chemical Lagrangian Model of the Stratosphere (CLaMS) constitute our case study. All datasets cover the period from June to August 2011 and are obtained from the Simulation Laboratory Climate Science at the Jülich Supercomputing Center and the Institute of Energy and Climate Research, Jülich, Germany (Griessbach et al., 2014). Fig. 3 presents a schematic diagram showing the data acquisition approaches followed in MIPAS and AIRS sensors. In addition, and for verification purposes, raw archived VIS (visible) and IR (infrared) images from geostationary weather satellites (3 h resolution) have been obtained from NCDC (National Climatic Data Center) at NOAA (National Oceanic and Atmospheric Administration). The following subsections describe briefly the four data sets.

4.1. MIPAS

The Michelson Interferometer for Passive Atmospheric Sounding (MIPAS) was operated by the European Space Agency from 2002 to 2012. The sensor measures the atmosphere tangentially (i.e., the beam is tangential to the Earth). The satellite finishes almost 14 orbits every day and the longitudes of orbits change from day to another. MIPAS data has very poor horizontal resolution since the measurements are taken along the vertical direction only (almost 3 km vertical resolution). The vertical extent is from 5 to 70 km height along the orbit track (note that the vertical profiles are not exactly vertical due to the satellite speed). The

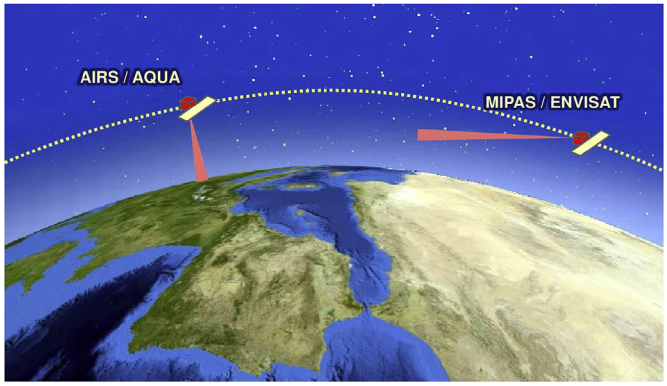


Fig. 3. Schematic diagram of the scanning methodologies of AIRS/AQUA and MIPAS/Envisat satellites. AIRS scans the region of the atmosphere directly below the satellite (at nadir) while MIPAS instrument scans the atmosphere tangential to the Earth. Note that CLaMS trajectories are only given for MIPAS detections (and not for AIRS).

advantages of MIPAS are that it provides very good vertical resolution and is highly sensitive in detecting aerosol concentrations. Fig. 4 a gives a daily sample full Earth scan of the satellite (i.e., 14 orbits) while Fig. 4 b shows a 3D depiction of a single orbital data track showing the resolution of the vertical profiles. The data set is time enabled point data with attributes containing time, longitude, latitude, altitude, orbit number, profile number, and detection codes. Detection codes are 0 for clear air, 1 for ice detections, 2 for ash detections and 4 for aerosol (SO_2) detection.

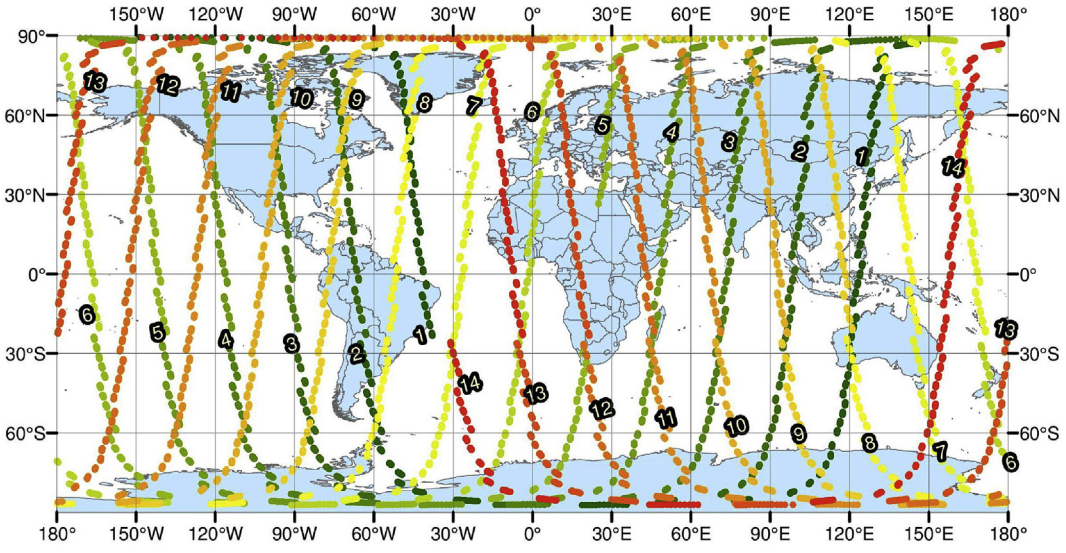
4.2. AIRS

The Atmospheric Infrared Sounder (AIRS) is loaded on NASA's Aqua orbital satellite, which was started in May 2002. AIRS detects the thermal energy of the atmospheric materials just below the satellite (i.e., nadir) and to both left and right of the ground track. A mirror rotates to scan the sides of the track. AIRS is characterized with good horizontal resolution and nearly global extent (twice a day). The disadvantages are the lack of elevation information, data gaps, and the different data acquisition times. AIRS performs 14.5 orbits every day and acquires almost 2.9 million IR measurements.

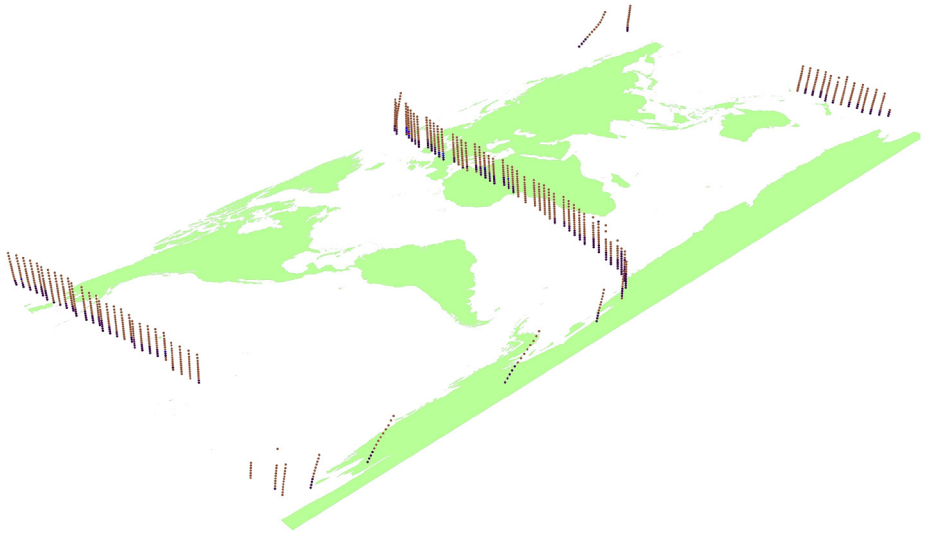
The data set contains two scalar values that are related to the sum of SO_2 and ash in the atmospheric column. The attributes include time, longitude, latitude, SO_2 index, ash index. Ash index means the existence of ash where negative values mean high concentrations. SO_2 index is related to the existence of SO_2 (high positive values mean high concentration of SO_2). The data is provided in files where each file represents a 12 h period (i.e., AM and PM for each day). The following query (Equation (5)) is used to discriminate significant detections out of the 1.4 million measurements in every 12 h data file:

$$(Ash < -0.5)\ \ AND\ \ (SO_2 > 1) \tag{5}$$

Fig. 5 shows a sample data file (i.e., 12 h) for AIRS data.



a



b

Fig. 4. a Sample full Earth tracks of MIPAS between 20110601010242 UTC and 20110602002534 UTC (i.e., almost during June 1st, 2011). The sequence of the tracks can be followed either by color or by following the track numbers shown on the figure. Note that the shown "dots" actually represents quasi-vertical profiles where each profile samples the atmosphere vertically every 3 km from 5 to 70 km altitudes (refer to Fig. 4b). The profiles are not exactly vertical due to the motion of the satellite. **b** 3D depiction of a single MIPAS track (track no. 6 in Fig. 4a) illustrating the vertical profiles of MIPAS data. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article).

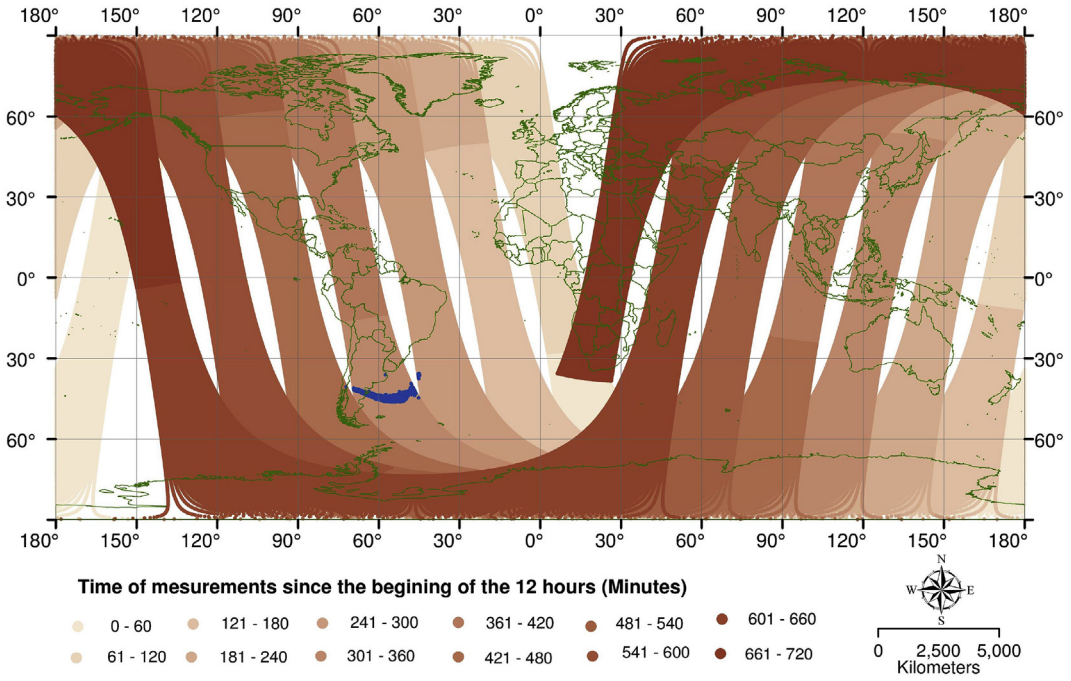


Fig. 5. Sample full Earth AIRS data file (containing 1.45 million point) during day 156 p.m. (June 5, 2011) between 20110605120531 UTC and 20110605235020 UTC. The legend shows the time of measurement during the 12 h period while the blue color depicts the significant detections (i.e., *Ash < -0.5 AND* $SO_2 > 1$) near Puyehue-Cordón eruption site. Note that AIRS data strips shift from data file to the next 12 h full Earth file and the time stamp also varies within the data file itself which create difficulties to convert the data to structured (i.e., raster) data formats. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article).

4.3. CLaMS

The Chemical Lagrangian Model of the Stratosphere (CLaMS) is a chemical transport model developed at Research Center Jülich, Germany. CLaMS was first described by McKenna et al. (2002) and was expanded into three dimensions by Konopka et al. (2007). It contains sub-modules for transport, chemistry, mixing and for microphysics that can be included or excluded. CLaMS dataset used in this research is developed using the trajectory module. This module determines the trajectories for individual air masses. The trajectories are included for MIPAS data set (described above). In other words, the data set is seeded at the locations of MIPAS detections where trajectories go back in time to the beginning of the data period, and extend in time for five consecutive days after the acquisition time of every detection.

The trajectories can be used to determine the spatial and temporal relations between the different data. In other words, for each MIPAS detection, CLaMS dataset has a single polyline. The polyline is seeded at the location of the MIPAS detection and extends forward and backward in time as described above. Points on the polyline are sorted in increasing temporal order. The raw data attributes contain the following fields per point: time, longitude, latitude, altitude, and the dynamical properties. The data set is pre-processed to add the following additional attributes: polyline number, a_1, a_2, b_1, b_2, c_1, c_2 (Section 2.1). Fig. 6 presents a sample trajectory that explains CLaMS data.

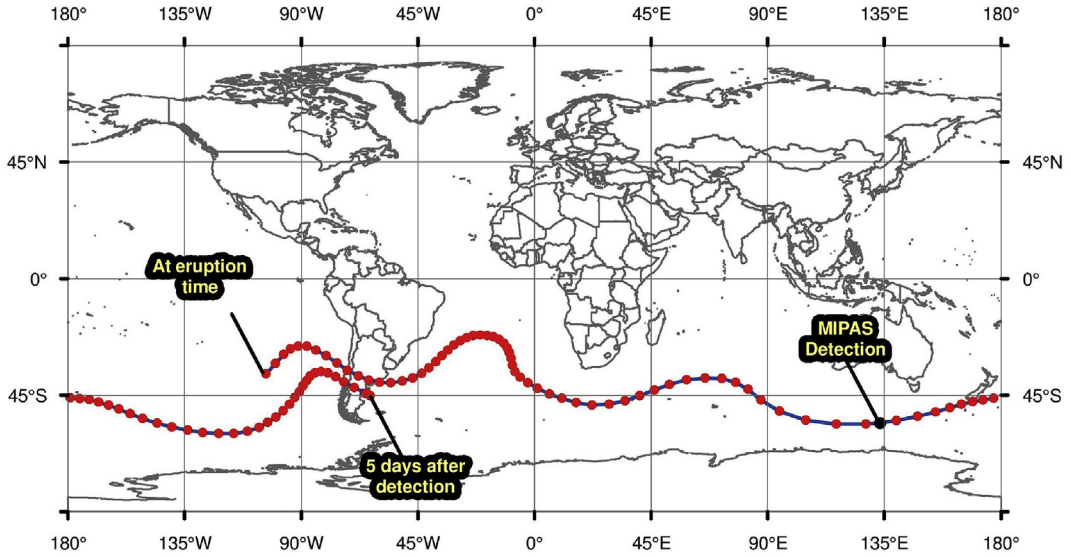


Fig. 6. Sample CLaMS trajectory corresponding to a MIPAS detection. The detection is shown as large dot south of Australia. The trajectory stretches back in time to the eruption time (June 4, 2011) and forward in time to five days after the time of the detection. The time step of CLaMS simulation (i.e. time step between the points on the trajectory) is 3 h.

4.4. Weather satellite images

The raw images are from seven geostationary satellites (GOES11, GOES12, GOES13, METEOSAT7, METEOSAT9, FEN YUNG 2E, and MTSAT). GIS (Geographic Information System) is used to pre-process the raw IR images as follows: *(1)* the raw images are first geo-referenced in the satellite vertical perspective projection; *(2)* the images are then re-projected in the common coordinate system (i.e., geographic WGS84 system); *(3)* finally a mosaic of the projected images is then produced in the common coordinate system to generate one image covering the globe at every time step. Fig. 7 a shows sample 2 full disk images from GOES12 and MTSAT geo-referenced each in its separate satellite perspective coordinate system while Fig. 7 b shows the same two images after re-projection in the common geographic coordinate system. It should be noted that care should be taken in discriminating volcanic ash plumes from convective clouds since they both appear similar on IR weather satellite images. Hence, all possible information (i.e., AIRS and MIPAS) can help to determine where to look for the ash plume on weather satellite mages.

5. Application

The following subsections demonstrate the effectiveness of the TTA at performing a number of tasks; namely, artificial plume generation, spatiotemporal correction, and the joining of multi-source spatiotemporal data.

5.1. Generation of CLaMS-based volcanic plumes

We first illustrate the TSTI effectiveness by generating CLaMS-based plume (i.e., simulation plume) for Puyehue-Cordón volcano on a high resolution time scale. The simulated plume can be used for assessing the

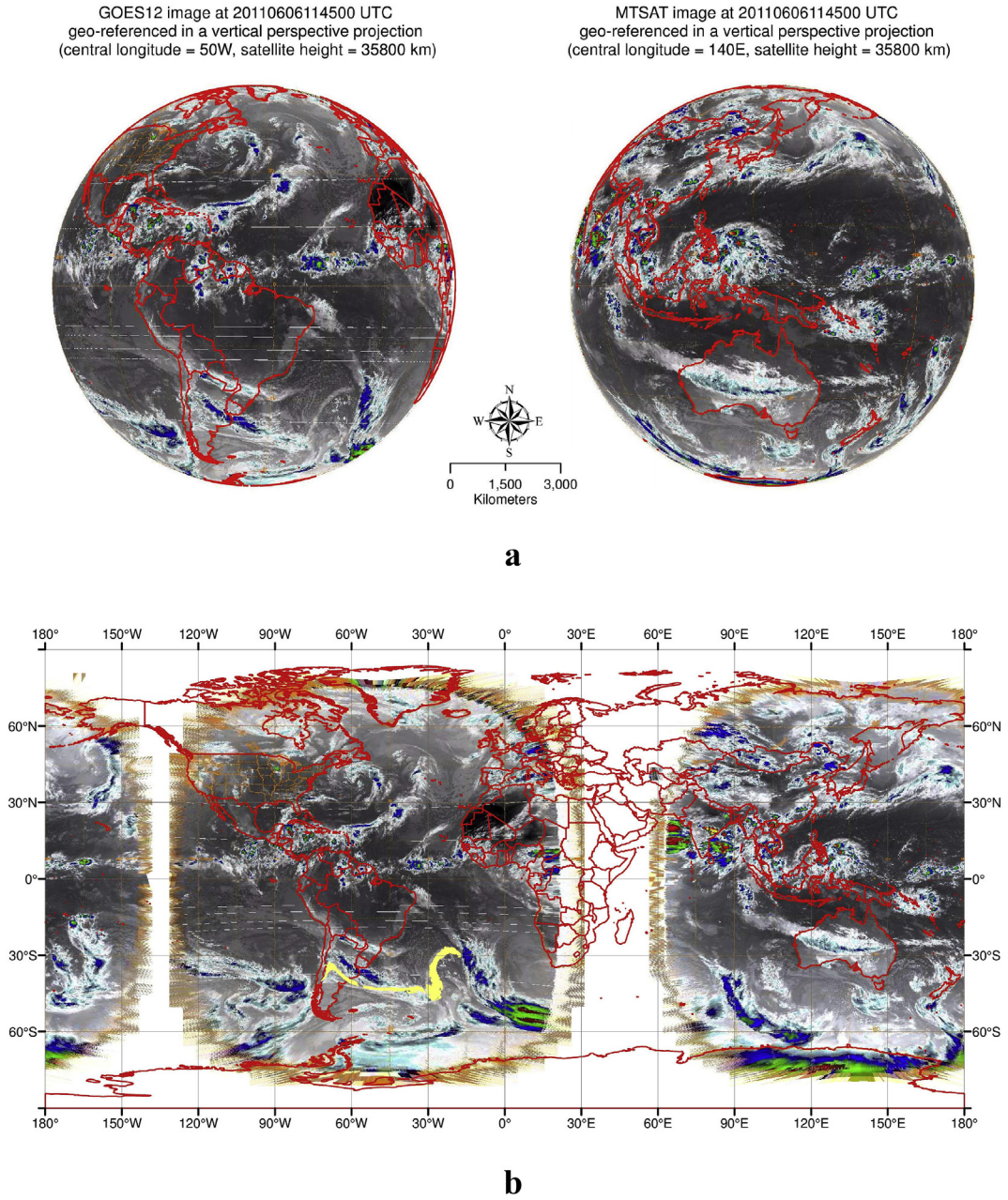


Fig. 7. a Sample 2 IR images from GOES12 and MTSAT at 20110606114500 UTC geo-referenced each in its separate vertical satellite perspective coordinate system. **b** The two sample IR images at 20110606114500 UTC after re-projection in the World Geodetic System 1984 (WGS 1984) common geographic coordinate system.

impact on aviation. It should be noted here that there are a group of different plume simulation models (e.g., Stohl et al., 2005; Stunder et al., 2007; Dacre and Coauthors, 2011). The performance of these models suffers complications related to the uncertainty of the initial condition as well the other problems related to the accuracy on the long term (Vernier et al., 2013). Complications aside, we do not seek to compete with these models, as we are not tackling the physical process sensu stricto in terms of mass balance and dispersion/diffusion in the ambient atmosphere. Rather, we employ TSTI for plume simulation in order to demonstrate the effectiveness of the method and to provide a relatively simple alternative for the simulation of plume dispersion in real or near-real time. We anticipate that this method will be of particular use in assessing the impact of volcanic plumes on aviation, for example.

For this purpose, a number of points are injected at the eruptive vent (17 points are used on two concentric circles at the eruptive vent every 15 min time step during the eruption time period) and left to be slide by the TSTI method within the flow field (i.e., left to self-adjust within the flow field via the TSTI method). The 17 points include 1 point at the center and 8 points on each of the two circles (i.e., equally separated by 45° radial angles). The TSTI is used to advect the simulated plume by determining the positions of the growing plume array of points every time step. Note that the simulated plume points essentially do not have corresponding trajectories in CLaMS dataset and here lies the importance of TSTI.

Fig. 8 shows sequence frames of the simulated plume overlaid on top of GOES12 images at 3-h intervals (i.e., the temporal resolution of GOES12 archived data). The visual comparisons of the simulated plume and weather satellite images enable accuracy verification of both TSTI and CLaMS simulations. As shown in Fig. 8, the evolution of the TSTI simulated plume matches GOES12 IR images. The average correlation coefficient between a rasterized version of the simulated plume and the observed plume is 0.58. More details of this comparison can be found in Elshehaly et al. (2015).

It must be noted that the above simulation procedure does not account for the lateral dispersion of the plume. The seed points being advected here (i.e., growing array of points) can be considered as the plume core that may disperse/diffuse laterally. In order to account for lateral dispersion, a modification of the above procedure is required. This modification requires adding a concentration attribute field to the seeded points (originally seeded at the volcano vent). All points are considered "parents" and are allowed to give birth to lateral "child" points (i.e., allowed to split laterally). Fractions of the parent concentration are then divided on the children according to the lateral dispersion coefficient. The children are then added to the plume growing array, get advected again with the flow, and can give birth to new children. The inclusion of

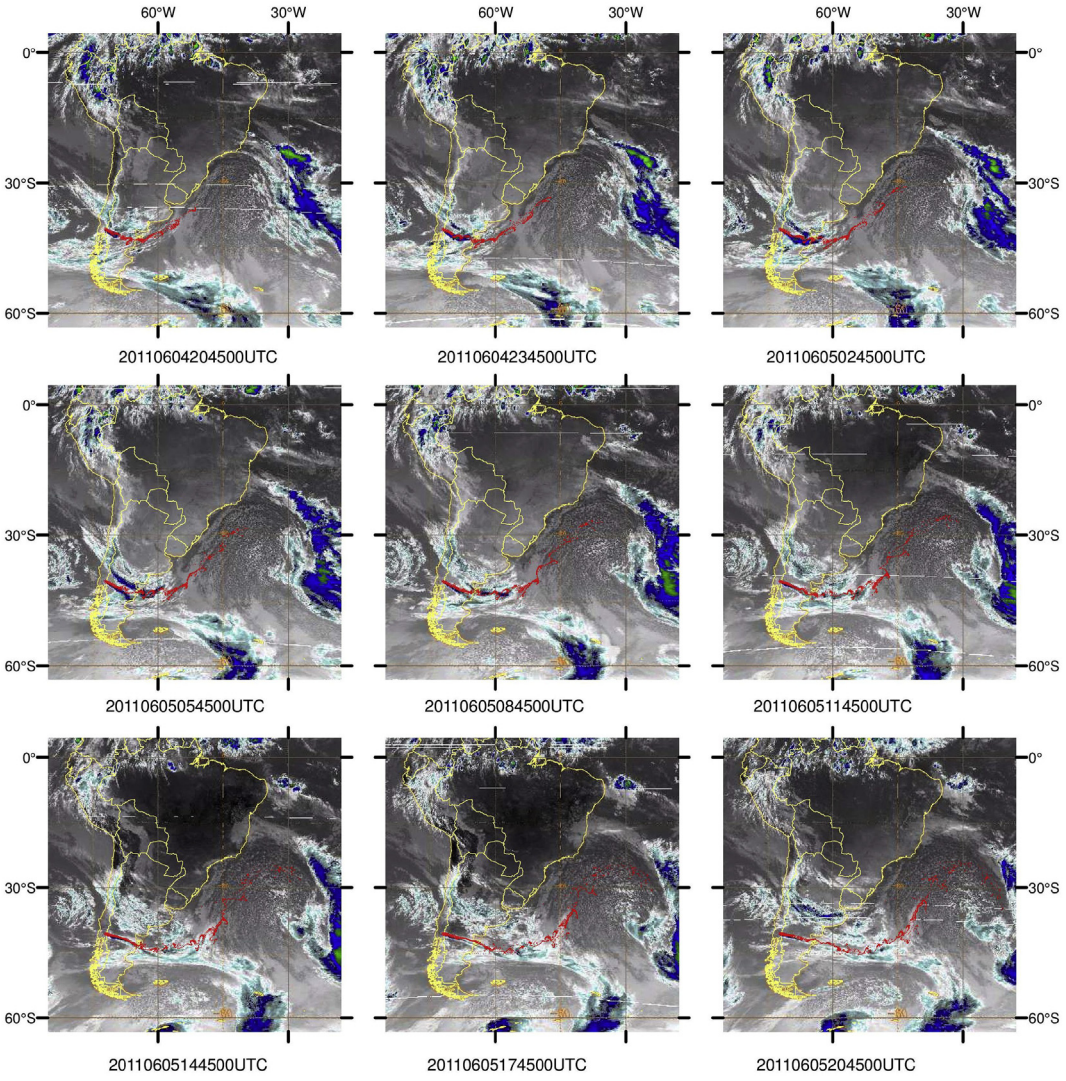


Fig. 8. TSTI simulated plume (shown in red dots) overlayed on top of GOES12 satellite images every 3 h time step (i.e., the time step of the archived satellites images). The plume is generated by injecting 17 points on the volcano opening every 15 min. At every time step, the newly injected points and the old plume points are advected using the developed TSTI method. Note that plume evolution perfectly matches the satellite images giving rise to the accuracies of both CLaMS simulation and the developed TSTI method. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article).

this lateral dispersion modification is left for future studies.

5.2. Construction of full earth AIRS scans (spatiotemporal correction)

The time stamps of MIPAS and AIRS data differ from one data point to another depending on the satellite position along its track. For example, an AIRS point may have been acquired in the beginning of the day while another point can be after 12 h and both are saved in the same data file. In addition, the difference in time between an AIRS strip and the adjacent strip is approximately 100 min (Fig. 5). Hence, this difference in time leads to discontinuity of the detected patterns since the patterns have actually been detected at different times.

For global presentation of the data or for the purpose of AIRS data rasterization, it is always desirable to construct a full Earth instantaneous scan of the data (i.e., shift the data to its location at the selected reporting time). For this purpose, TSTI is used to first project AIRS detections on MIPAS surfaces created exactly at the time of each AIRS detection (refer to the following section) then advect AIRS detections to any target time by interpolating its position at the required target time. Fig. 9 **a** presents a sample raw AIRS data where each detection has different acquisition time (i.e., "as is") while Fig. 9 b presents the same data after time-shift correction.

5.3. Spatiotemporal join between AIRS & MIPAS

MIPAS and AIRS are different sensors with individual strengths and weaknesses. In particular, AIRS provides contiguous satellite coverage whereas MIPAS data consists of far fewer points along tracks only. MIPAS provides solid altitude information while AIRS doesn't. It would be very useful to blend the data acquired from the two sensors. For example, if altitude information can be interpolated from MIPAS into AIRS (which is very difficult to be achieved using the traditional techniques of seeding data into the circulation model itself), this will blend the best of both worlds. This will be very useful to different applications such as evaluating how far the two instruments agree, identifying safe air corridors for flight traffic (i.e., in the horizontal and vertical), and for 3D plume visualization purposes. This may also prove extremely useful for modelling ash aggregation and disaggregation processes. See Mueller et al. (2017) for an example.

TSTI is used here also to perform the spatiotemporal interpolation from MIPAS altitudes into AIRS in order to assign elevation values to AIRS (Fig. 10). This assignment can be done either on-the-fly during time-shift correction (Section 4.3) or by processing the raw AIRS data directly. For every AIRS detection, the nearest 8 neighbors (in space and time) from CLaMS positions (that actually represent MIPAS detections) are selected and the spatiotemporal interpolation is performed.

6. GPU-BASED interactive tracking

The TTA (Trajectory-based Tracking Analysis) is amenable to massive parallelization, since it operates independently for every point that needs to be advected through the flow field. We have developed a GPU (Graphic Processing Unit)-based version of our tracking technique to

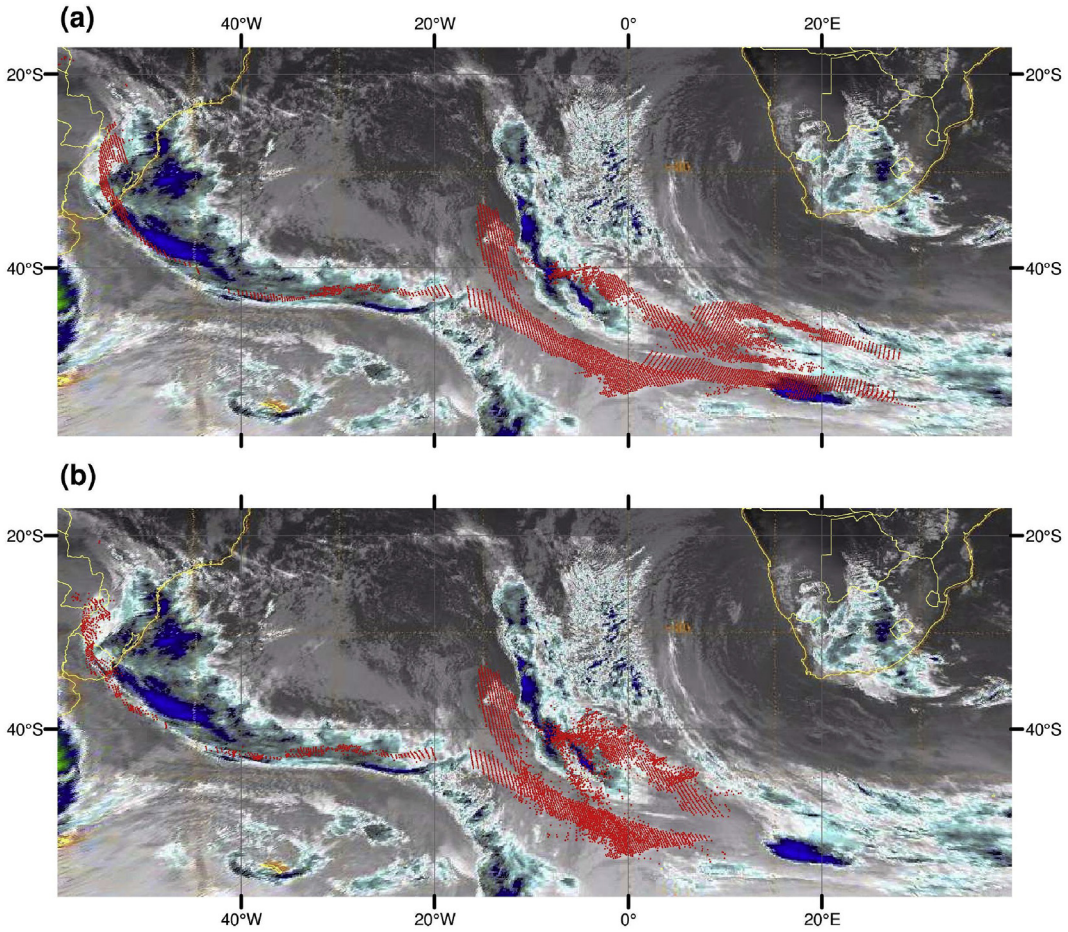


Fig. 9. Application of the TSTI method to spatiotemporally correct AIRS data. The upper panel shows sample raw AIRS data spanning 12 h "as is" and the lower panel shows the corrected AIRS data using the TSTI method. Note that the raw data have different acquisition times. The time difference between the patterns on the right and left of the raw data is almost 12 h. This is because the full Earth orbits started on the left and ended at the right having almost 14 orbits and 12 h in between. During this time the patterns on the left moved eastward and got detected again in the last orbit at the end of the full Earth coverage. Hence, parts of the patterns got detected twice and included in the same data file. The TSTI solved this problem by shifting all AIRS detections to its correct position at a target time equal to GOES 12 acquisition time and hence creating proper overlapping.

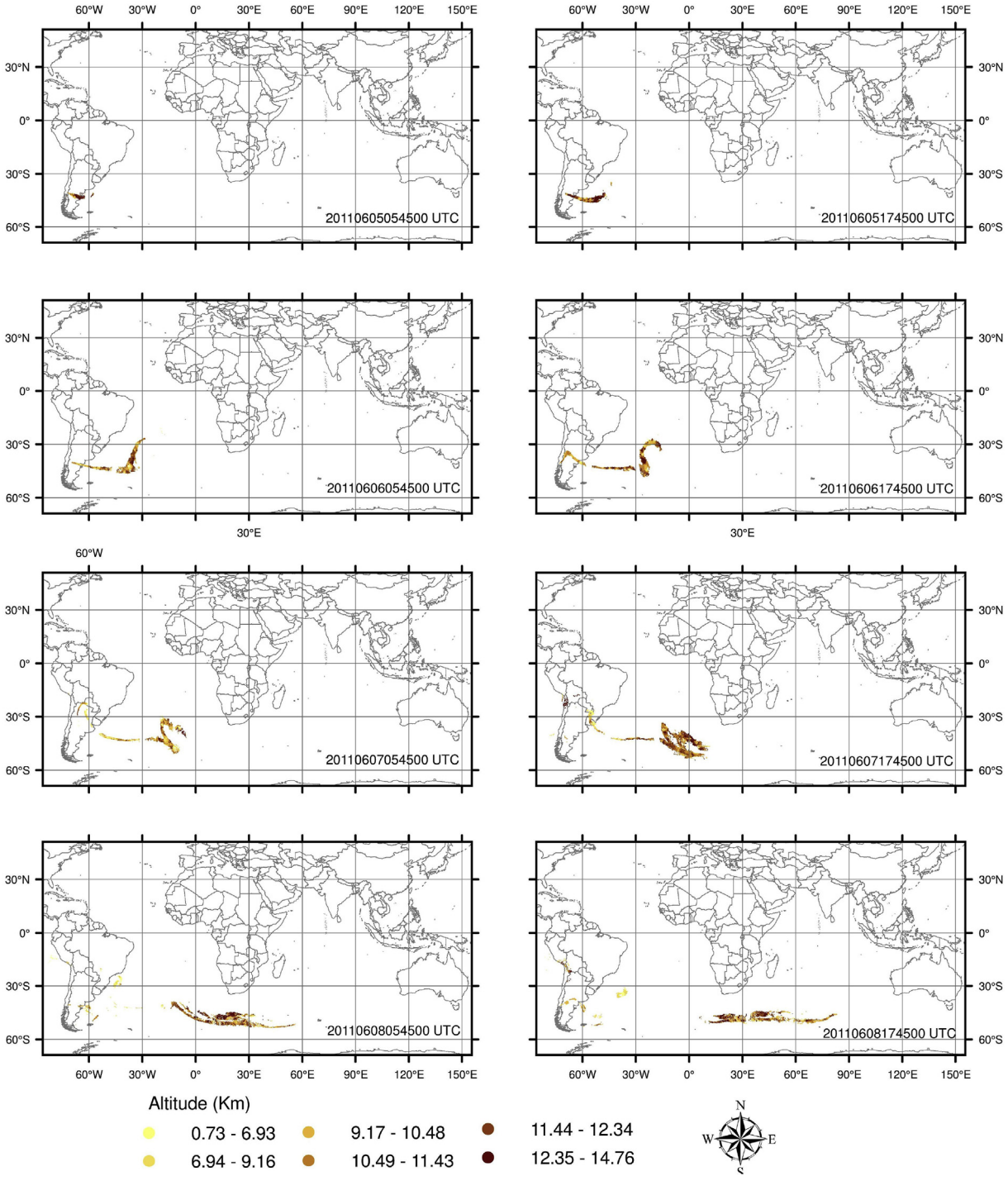


Fig. 10. Application of the TSTI to produce full Earth instantaneous AIRS scans at any selected target times (every 12 h is used here) and spatiotemporally join MIPAS altitudes to AIRS detection. The figure depicts the evolution of the 2011 Puyehue-Cordón volcanic plume as observed on TSTI-corrected AIRS data every 12 h. Note that AIRS altitudes have been assigned from MIPAS altitudes based on the trajectory-based spatiotemporal interpolation method (TSTI) developed in this research.

support interactive visualization for different purposes including the determination of particle sources (Elshehaly et al., 2014) and for creating region-of-interest reference datasets for the interactive data fusion and tracking (Elshehaly et al., 2015). To this end, the TSTI method is used during pre-processing to construct a regular reference model (i.e., simulated plume dataset) that densely represents the spatiotemporal evolution of the plume behavior in a Lagrangian (i.e., moving and growing) region of interest.

The reference model is then used in our GPU parallel algorithm to interactively advect AIRS measurements on the model and track their location at any given point in time. This allows the user to interactively correct AIRS detections at any time by controlling a time slider. In this case, the source time step *Ts* is determined by the timestamp of the original detection point while the time instance selected by the user becomes our destination time step *Td*. AIRS points are mapped to their nearest neighborhood in the CLaMS dataset using OpenGL's vertex shader, i.e. one detection point per shader instance. The use of a regular reference model, as constructed by TTA, optimizes global memory access on the GPU and yields interactive rates.

Table 1
TTA runtimes for 1.4 million CLaMS vertices (CPU).

[a]Sample size n	[b]No. Nbrhd used m	[c]Runtime (sec)
5000	8	7
1458000	8	1130

[a] Sample size n is the number of points requiring spatiotemporal interpolation.
[b] No. Nbrhd used *m is the number of nearest trajectories used in the spatiotemporal interpolation*.
[c] Runtime on Core 2Duo @ 2.66 Ghz, 3.5 GB RAM (including the time to load the simulation data into memory).

7. Technical details

Due to the large storage size of the data involved, high level programming environments were not feasible since they would consume the computation resources and take too long to process the data. Accordingly, plain C programming was used to implement TSTI. The main idea is that CLaMS data is read into memory as a two-dimensional array in which the first index is the polyline number and the second is time. Hence, CLaMS data is quickly retrieved for any trajectory at any desired time. CPU runtime is generally within few seconds to a few tens of minutes (Table 1). This is quite acceptable for offline preprocessing. However, this runtime can be significantly improved using acceleration data structures to speed up the neighborhood search within the simulation data set (i.e., CLaMS).

We rendered the visualizations using OpenGL 4.4 on an NVIDIA GeForce GT 740 M/PCIe/SSE2 GPU, and Shader Language GLSL 4.40. The developed GPU interface enabled motion animation of all overlaid datasets at interactive rates. In addition, the results of the CPU-TSTI simulations were fed to the shaders in the vertex array buffer and were rendered at an average frame rate of 60 FPS. Color encoding was controlled through attribute sliders that enable the user to interactively modify the visual encoding of the plume points and display different value combinations of SO_2 and ash index attributes (refer to Equation (5)). The values set by the user through the sliders were passed to the shader as uniforms and were used as parameters to the calculation of both color and opacity for each individual point.

In our GPU-based tracking, raw AIRS data (i.e. original detection points) was fed to the vertex shader. The shader then performed the nearest neighbor search and interpolation calculations in parallel on each point. The average frame rate achieved for both tracking and rendering is 20 FPS. This frame rate can be improved through the use of a more sophisticated nearest neighbor search, which is the focus of our future work. The details of the GPU implementation can be found in Elshehaly et al. (2015).

8. Conclusions and discussion

This paper presents a spatiotemporal interpolation technique TSTI (Trajectory-based Spatial and Temporal Interpolation) for spatiotemporal interpolation of time variant data in unsteady flow field. The effectiveness is demonstrated using multiple source data sets for eruption events in the atmosphere. The demonstration included different tasks such as plume simulation, spatiotemporal joining of data from multiple sources, spatiotemporal correction, re-sampling of irregular space-time data into regular datasets, filling data gaps, and the generation for region-of-interest reference datasets for GPU-based visualizations. The TSTI showed excellent results and can be considered (to the best of the authors' knowledge) the first efficiently automated geostatistical technique to advect/join large scatter and time-enabled data in space and time.

The main application that strongly benefits from the TSTI method is the construction of global instantaneous maps of polar satellite data and joining this data to measurements acquired by other sensors regardless of the variable being measured. The variable under consideration can be a volcanic substance, pollutant, smoke, cloud, or even rainfall intensity. Indeed, the construction of instantaneous global rainfall intensity mosaics from the time variant strips of polar weather satellites, in concert with data from other sensors, is another strong candidate for the application of our method. In general, the TSTI method may be applied to any problem involving observations of a spatiotemporally-variable substance freely moving in a flow field (especially in large-scale atmospheric and oceanic circulations).

The accuracy depends mainly on the accuracy of the underlying trajectory information describing the flow field. For the case study described in this paper, an alternative for the underlying numerical simulation is the flow fields developed through the known techniques of cross-correlating portions of geostationary weather satellite images in the successive times. Ongoing research includes the conversion of TSTI into real-time forecasting mode by using exponential smoothing to encode the method.

References

Bignami, C., Corradini, S., Merucci, L., De Michele, M., Raucoules, D., De Astis, G., Stramondo, S., Piedra, J., 2014. Multisensor satellite monitoring of the 2011 Puyehue-Cordon Caulle eruption. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 7 (7), 2786–2796.

Burger, Kai, Ferstl, Florian, Theisel, Holger, Westermann, Rüdiger, 2009. Interactive streak surface visualization on the GPU. Vis. Comput. Graph. IEEE Trans. 15 (6), 1259–1266.

Castro, J.M., Schipper, C.I., Mueller, S.P., Militzer, A.S., Amigo, A., Parejas, C.S., Jacob, D., 2013. Storage and eruption of near-liquidus rhyolite magma at Cordon Caulle, Chile. Bull. Volcanol. 75 (4), 117.

Collini, E., Osores, M.S., Folch, A., Viramonte, J.G., Villarosa, G., Salmuni, G., 2013. Volcanic ash forecast during the June 2011 cordon Caulle eruption. Nat. Hazards 66 (2), 389–412.

Dacre, H.F., Coauthors, 2011. Evaluating the structure and magnitude of the ash plume during the initial phase of the 2010 Eyjafjallajє́okull eruption using lidar observations and NAME simulations. J. Geophys. Res. 116 https://doi.org/10.1029/2011JD015608. D00U03.

Elshehaly, M., Gracanin, D., Gad, M., Elmongui, H.G., Matkovic, K., 2014. Poster: "StreamProbe: a novel GPU-based selection technique for interactive flow field exploration. In: Proceeding of the IEEE VIS 2014 Conference, Paris, 9–14 Nov.

Elshehaly, M., Gracanin, D., Gad, M., Elmongui, H.G., Matkovic, K., 2015. Interactive Fusion and Tracking for Multi-modal Spatial Data Visualization. Eurographics Conference on Visualization (EuroVis), Italy.

Fairlie, T.D., Vernier, J.-P., Natarajan, M., Bedka, K.M., 2014. Dispersion of the Nabro volcanic plume and its relation to the Asian summer monsoon. Atmos. Chem. Phys. 14, 7045–7057. https://doi.org/10.5194/acp-14-7045-2014, 2014.

Franke, R., 1982. Smooth interpolation of scattered data by local thin plate splines. Comput. Math. Appl. 8 (4), 237–281.

Griessbach, S., Hoffmann, L., Spang, R., Riese, M., 2014. Volcanic ash detection with infrared limb sounding: MIPAS observations and radiative transfer simulations. Atmos. Meas. Tech. 7, 1487–1507.

Isaaks, E., Srivastava, R., 1989. Applied Geostatistics. Oxford University Press, p. 542.

Kluser, L., Erbertseder, T., Meyer-Arnek, J., 2013. Observation of volcanic ash from puyehue-cordon Caulle with IASI. Atmos. Meas. Tech. 6 (1), 35.

Kohrs, R.A., Lazzara, M.A., Robaidek, J.O., Santek, D.A., Knuth, S.L., 2013. Global satellite composites—20 years of evolution. Atmos. Res. 135–136, 8–34. https://doi.org/10.1016/j.atmosres.2013.07.023.

Konopka, P., Günther, G., Müller, R., dos Santos, F.H.S., Schiller, C., Ravegnani, F., Ulanovsky, A., Schlager, H., Volk, C.M., Viciani, S., Pan, L.L., McKenna, D.-S., Riese, M., 2007. Contribution of mixing to upward transport across the tropical tropopause layer (TTL). Atmos. Chem. Phys. 7, 3285–3308.

Kruger, Jens, Kipfer, Peter, Konclratieva, P., Westermann, Rüdiger, 2005. A particle system for interactive visualization of 3D flows. Vis. Comput. Graph. IEEE Trans. 11 (6), 744–756.

Lane, D.A., 1996. Visualizing Time-varying Phenomena in Numerical Simulations of Unsteady Flows. NASA Ames Research Center. February 1996.

Max, N., Becker, B., 1999. Flow visualization using moving textures. In: Wiley, John, Sons, Ltd (Eds.), Data Visualization Techniques. Chandrajit Bajaj, pp. 99–105.

McKenna, D.S., Konopka, P., Grooß, J.-U., Günther, G., Müller, R., Spang, R., Offerman, D., Orsolini, Y., 2002. A new chemical lagrangian model of the stratosphere (ClaMS), 1, formulation of advection and mixing. J. Geophys. Res. 107 (D16).

McLoughlin, Tony, Laramee, Robert S., Peikert, Ronald, Post, Frits H., Chen, Min, 2010. Over two decades of integration-based geometric flow visualization. In: Computer Graphics Forum, 29(6). Blackwell Publishing Ltd, pp. 1807–1829, 2010.

Montmollin, F.A., Olivier, R.J., Simard, R.G., Zwahlen, F., 1980. Evaluation of a precipitation map using a smoothed elevation–precipitation relationship and optimal estimates (kriging). Nord. Hydrol. 11 (3–4), 113–120.

Mueller, S.B., Ayris, P.M., Wadsworth, F.B., Kueppers, U., Casas, A.S., Delmelle, P., Taddeucci, J., Jacob, M., Dingwell, D.B., 2017. Ash aggregation enhanced by deposition and redistribution of salt on the surface of volcanic ash in eruption plumes. Sci. Rep. 7, 45762.

Philip, G.M., Watson, D.F., 1982. A precise method for determining contoured surfaces. Aust. Petrol. Explor. Assoc. J. 22, 205–212.

Post, Frits H., Vrolijk, Benjamin, Hauser, Helwig, Laramee, Robert S., Doleisch, Helmut, 2003. The state of the art in flow visualization: feature extraction and tracking. In: Computer Graphics Forum, 22(4). Blackwell Publishing, Inc, pp. 775–792, 2003.

Schafhitzel, Tobias, Tejada, Eduardo, Weiskopf, Daniel, Ertl, Thomas, 2007. Point-based stream surfaces and path surfaces. In: Proceedings of Graphics Interface 2007. ACM, pp. 289–296, 2007.

Schipper, C.I., Castro, J.M., Tu_en, H., James, M.R., How, P., 2013. Shallow vent architecture during hybrid explosive-effusive activity at Cordon Caulle (Chile, 201112): evidence from direct observations and pyroclast textures. J. Volcanol. Geotherm. Res. 262, 25–37.

Stohl, A., Forster, C., Frank, A., Seibert, P., Wotawa, G., 2005. The Lagrangian particle dispersion model FLEXPART version 6.2. Atmos. Chem. Phys. 5, 2461–2474. https://doi.org/10.5194/acp-5-2461-2005.

Stunder, B.J.B., Heffter, J.L., Draxler, R.R., 2007. Airborne volcanic ash forecast area reliability. Weather Forecast. 22, 1132–1139.

Tuffen, H., James, M.R., Castro, J.M., Schipper, C.I., 2013. Exceptional mobility of an advancing rhyolitic obsidian flow at Cordon Caulle volcano in Chile. Nat. Commun. 4.

Vernier, J.P., Fairlie, T.D., Murray, J.J., Tupper, A., Trepte, C., Winker, D., Pelon, J., Garnier, A., Jumelet, J., Pavolonis, M., Omar, A.H., 2013. An advanced system to monitor the 3D structure of di_use volcanic ash clouds. J. Appl. Meteorol. Climatol. 52 (9), 2125–2138.

Wu, X., Griessbach, S., Hoffmann, L., 2017. Equatorward dispersion of high-latitude volcanic plume and its relation to the Asian summer monsoon: a case study of the Sarychev eruption in 2009. Atmos. Chem. Phys. Discuss. https://doi.org/10.5194/acp-2017-425, in review, 2017.