



Determining Real-Time Patterns of Lightning Strikes from Sensor Observations

Bolelang H. Sibolla^{1,2} · Terence Van Zyl³ · Serena Coetzee²

Accepted: 10 November 2020
© The Author(s) 2021

Abstract

Transient spatiotemporal events occur within a short interval of time, in a particular location. If such events occur unexpectedly with varying durations, frequencies, and intensities, they pose a challenge for near-real-time monitoring. Lightning strikes are examples of such events and they can have severe negative consequences, such as fires, or they precede sudden flash storms, which can result in damage to infrastructure, loss of Internet connectivity, interruption of electrical power supply, and loss of life or property. Furthermore, they are unexpected, momentary in occurrence, sometimes with high frequency and then again with long intervals between them, their intensity varies considerably, and they are difficult to trace once they have occurred. Despite their unpredictable and irregular nature, timely analysis of lightning events is crucial for understanding their patterns and behaviour so that any adverse effects can be mitigated. However, near-real-time monitoring of unexpected and irregular transient events presents technical challenges for their analysis and visualisation. This paper demonstrates an approach for overcoming some of the challenges by clustering and visualising data streams with information about lightning events during thunderstorms, in real time. The contribution is twofold. Firstly, we detect clusters in dynamic spatiotemporal lightning events based on space, time, and attributes, using graph theory, that is adaptive and does not prescribe number and size of clusters beforehand, and allows for use of multiple clustering criteria and thresholds, and formation of different cluster shapes. Secondly, we demonstrate how the space time cube can be used to visualise unexpected and irregular transient events. Along with the visualisation, we identify the interactive elements required to counter challenges related to visualising unexpected and irregular transient events through space time cubes.

Keywords Transient events · Spatiotemporal event clustering · Streaming data · Visual analytics · Space time cube · Lightning data

✉ Bolelang H. Sibolla
bsibolla@csir.co.za

Terence L. Van Zyl
tvanzyl@uj.ac.za

Serena Coetzee
serena.coetzee@up.ac.za

- ¹ Council for Scientific and Industrial Research, Pretoria, South Africa
- ² Centre for Geoinformation Science, Department of Geography, Geoinformatics and Meteorology, University of Pretoria, Pretoria, South Africa
- ³ Institute for Intelligent Systems, University of Johannesburg, Johannesburg, South Africa

Introduction

An event occurs in a certain place during a particular interval of time and is characterised by its start and end times, its spatial coordinates, and observed descriptive attributes (Kisilevich et al. 2010; Andrienko et al. 2015). In this paper, we focus on transient spatiotemporal events that occur unexpectedly and within a very short interval of time, such as within a few seconds or less. Their duration varies and is unpredictable, and their frequencies and intensities vary considerably. An example of such events are lightning strikes, which can lead to a number of negative consequences, such as fires, preceding sudden flash storms, resulting in damage to infrastructure, loss of Internet connectivity, interruption of electricity power supply, and loss of life or property (Podur et al. 2003; Iordanidou

et al. 2016; Wu et al. 2017; Mofokeng et al. 2019). The challenge presented by lightning strikes is their unexpected, momentary occurrence with considerable variation in their frequency and intensity, and the difficulty to trace once they have occurred. Despite their unpredictable nature, timely analysis thereof is crucial in order to understand the patterns and behaviour so that adverse effects may be mitigated.

Transient spatiotemporal events, as observations of a phenomena, are often captured by sensors. Information about them may be transmitted by the sensing devices, typically as streams of data with the expectation that they will either be processed immediately or stored in a database for analysis later. Unexpected and irregular transient events pose a number of technical challenges for real-time monitoring related to the analysis and effective visualisation of the streams of data with information about them.

This paper describes an implementation of real-time analysis and sense making of transient events by a human expert. Determining lightning clusters during a storm is used as a case study. Clustering of lightning strikes assists users to understand and assess areas that have been affected by lightning over time. Due to the high frequency of lightning strikes, clustering is necessary to enable the analysis of the resulting patterns. The clustering of lightning events in real-time requires analysis of multiple aspects of the lightning event. We achieve clustering of the lightning events by a graph-based clustering method using a connected components algorithm. Apart from the efficiency in retrieval of the clusters, the advantage of the graph-based methods lies in its ability to support various node connectivity relationships using multiple edge attributes. This is important for lightning observations that consist of various attributes such as type of lightning strike, intensity of the strike, make and model of the sensor network the observations were acquired from, and many others. The algorithm presents an extensible clustering method that can support multiple relationship attributes between real-time events. This graph-based method provides a framework for adding events and event attributes in real time, to be analysed and visualised. We follow the visual analytics approach of using geospatial techniques for the analysis and aiding sense making through interactive visualisations, while allowing the user to influence the analysis process, as and when required, based on their domain knowledge (Keim et al. 2008; Andrienko et al. 2010).

This study had two objectives. Firstly, we show how graph theory can be used to detect clusters in transient, unexpected, and highly dynamic spatiotemporal events, using space, time, and multiple event attributes. Secondly, we demonstrate an application of the Space Time Cube as a dynamic spatiotemporal visualisation environment, for visualising unexpected and irregular transient events. As part of the second objective, we also identify the interactive

elements required to counter the analysis and visualisation challenges that affect this type of data.

The paper is organised as follows: Section “[Related Work](#)” provides a review of literature with respect to the various clustering topics, and visual analytics of event data. Section “[Method](#)” details the graph-based spatiotemporal event clustering methodology developed in this study. Section “[Visualisation and Interaction](#)” illustrates how visual analytics was used to provide insights into real-time lightning clustering patterns. Section “[Visualisation Results](#)” presents the results of lightning clusters that were detected in real-time over different periods. Section “[Discussion](#)” discusses the merits and disadvantages of methods developed in this study, while mentioning recommendations of future work.

Related Work

In order to understand the different aspects applied in this work, three topics are reviewed, namely spatiotemporal event clustering in Section “[Spatiotemporal Data Clustering](#)”, lightning event clustering in Section “[Lightning Event Clustering](#)”, and streaming data clustering in Section “[Streaming Data Clustering](#)”. These sections lay the groundwork for the work done to meet the first objective of detecting clustering patterns in transient, unexpected, spatiotemporal events with highly dynamic attribute values and variable data frequency. Following the review of various clustering methods, we present an argument for the method applied in this section that details the similarities and contributions to existing methods. We further review state-of-the-art methods and open research challenges in the way visual analytics of transient events has been applied and highlight the challenges, relative to this study, in order to address the second objective of this research.

Spatiotemporal Data Clustering

Clustering of spatiotemporal data is a method of assessing similarities in data with the purpose of finding new and interesting patterns. These groupings of data can be defined by attribute similarity, and spatial and temporal proximity, resulting in detection of different cluster types. In recent literature, the most widely used classification of clustering methods is that by Han et al. (2011), who classify clustering methods into four categories, namely partitioning, hierarchical, density-based, and grid-based methods. In clustering methods, the main difference between spatial and spatiotemporal clustering is the introduction of the time variable. The time variable can be treated as either an attribute or a dimension. Where time is treated as a dimension, for example observations recorded as $[(X, Y, T) + \textit{attribute}]$, spatiotemporal clustering

methods have been developed based on the existing spatial clustering methods (Shi and Pun-Cheng 2019). We treat the time element of spatiotemporal lightning event sensor observations as a third dimension; therefore, extension of spatial clustering methods is considered.

Two recent reviews of spatiotemporal data clustering are found in literature, namely by Ansari et al. (2019) and Shi and Pun-Cheng (2019). Ansari et al. (2019) present a comprehensive review of spatiotemporal clustering which begins with a taxonomy of spatiotemporal data types, followed by a discussion of clustering approaches, spatiotemporal patterns, and software tools available. The taxonomy classifies spatiotemporal data types as events, geo-referenced data items, geo-referenced time series, moving objects, and trajectories. The clustering approaches are discussed in relation to the data types presented in the taxonomy. Of particular interest to this study, from Ansari et al. (2019) review, is clustering of events. According to literature, individual event data items have no identification (Tork 2012); as a result, events can not be tracked or accessed by their identity over time, in clustering methods. Therefore, event clustering focuses on discovering groups of events that occur close to each other in space and time, and also share similar attributes (Kisilevich et al. 2010). Examples of event clustering algorithms listed by Ansari et al. (2019) include ST-GRID, Fuzzy C-Means, space-time scan statistics, and ST-DBSCAN. ST-GRID partitions spatial and temporal dimensions into gridded cells, where the grid cell size is determined using the $k - dist$ graph. The event points within a cell are counted and if the number of points is equal to or greater than $k + 1$, then the cells are merged as a cluster. ST-DBSCAN is an extension of the density-based clustering method, DBSCAN and differentiates between core points, noise, and adjacent clusters. It makes use of two parameters, spatial neighbourhood radius, and temporal neighbourhood radius, which are both determined using the $k - dist$ graph, and a core point is determined if the number of points in its neighbourhood is equal to or greater than the threshold. Space time scan statistics is based on the scanning window process where a cylindrical scanning window is defined where an appropriate radius is selected to detect clusters of significant size based on a statistical significance test. Spatiotemporal Extended Fuzzy C-Means makes use of a distance function that considers both spatial and temporal dimensions by means of a multiplicative parameter λ .

While Ansari et al. (2019) taxonomy classifies data types as described above, Shi and Pun-Cheng (2019) taxonomy of data types refers to the fundamental types of spatiotemporal data; points, lines, and polygons. Although cognisant of these other data types in the taxonomy, Shi and Pun-Cheng (2019) review only addresses clustering of point-based data, therefore excluding topics such as trajectory

clustering, which are mentioned by Ansari et al. (2019). However, this review is relevant to this study because lightning event observations are point based. Shi and Pun-Cheng (2019) divide existing clustering methods into hypothesis testing-based methods and partitional clustering methods. Hypothesis testing-based methods originate from the field of statistics, where hypothesis testing is used to determine the probability that a given hypothesis is true or false. These methods include space-time interaction methods, spatiotemporal k -nearest Neighbour (KNN) test, and scan statistics. Space time interaction methods assess the interaction between the space and time dimensions separately. An example of such an application is the Knox method, where the critical space and time distances respectively are manually defined beforehand, and pairs of data are assessed for nearness based on the critical distance. Spatiotemporal KNN tests assess the k nearest neighbour in space and time simultaneously. Based on the concept of a scan window, this method uses a circular scan window with different radii to find circular clusters of two-dimensional spatial data with a statistical significance test where the upper limit of the circle should not include more than 50% of all the dataset, normally. Each point could be the centre of a circular scan window that contains different numbers of other points. In 3-dimensional space where time is the third dimension, the circle becomes a cylindrical scan window. Partitional clustering methods use distance functions to determine whether a point belongs to a cluster or noise. These include density-based methods, such as DBSCAN, kernel density estimation, and windowed nearest neighbour.

We note that Ansari et al. (2019) and Shi and Pun-Cheng (2019) report on similar clustering approaches for event data. The former mentions applications and algorithms that use these approaches, whereas the latter classifies these existing approaches and provides examples. Both reviews agree that no one method of clustering is suitable for all data types. The role of user-specific parameters to the clustering process is thus vital. The ability to adjust parameters of the clustering method is required to archive optimum results. Shi and Pun-Cheng (2019) further state that apart from the need for development of new clustering algorithms, there is also a need for predefined thresholds (e.g. distances, radius, and density) based on expert knowledge. These conclusions are discussed in support of the clustering method applied in this study later in Section “[Relevance of Related Work to Event Clustering](#)”.

Lightning Event Clustering

Lightning strikes are classified as events and more particularly transient events. The clustering behaviour of lightning strikes has been studied by many, for example for the purpose of determining the long-term behaviour

of lightning strikes in a specific region (Mofokeng et al. 2019; Podur et al. 2003; Iordanidou et al. 2016), or for the purpose of now-casting for early warning and alerting purposes (Kohn et al. 2011; Strauss et al. 2013). This paper is concerned with real-time analysis for early warning and alerting purposes. The existing methods of determining clusters of lightning events make use of spatiotemporal partitioning methods of point-based event data as discussed by Shi and Pun-Cheng (2019). We discuss two examples based on the work of Podur et al. (2003) and Mofokeng et al. (2019).

Podur et al. (2003) review a two-decade-long time series of retrospective lightning events to determine clustering events that lead to wildfires. They used a nearest-neighbour statistic, K-function, to test whether the events are random, clustered, or regular; and the Kernel density estimator to analyse the spatial intensity of the events. Iordanidou et al. (2016) use k-means analysis to determine the clusters of lightning points in space and time. In their method, they start with k random initial points as centres of the clusters and assign the points to the closest clusters. Subsequently, the centres of the clusters are updated to be the mean of the constituent points. They decide the number of clusters using G-means which splits the data into groups until the data is assigned to each cluster following a Gaussian distribution. Mofokeng et al. (2019) used average nearest neighbour (ANN) analysis to determine the spatial clustering pattern of lightning events.

The studies above all show clustering on retrospective data. In the case of near-real-time clustering, two main studies were identified. Kohn et al. (2011) discuss dynamic clustering in near-real-time, using K-means for now-casting of lightning events, for purposes of early warning detection. Strauss et al. (2013) proposed the use of a kernel density estimator on a temporal sliding window on lightning data for detection and tracking of active cells. In their method, Strauss et al. (2013) apply a fixed-width temporal window, which slides in time with a constant rate, for discrete time-steps, in order to screen for new incoming lightning events.

The methods applied to lightning events are consistent with the partitioning approach of spatiotemporal events as described by Shi and Pun-Cheng (2019). In addition, the sliding window approach is consistent with analysis of sequential data and temporally dynamic phenomena, whereas the size of the window may be defined in varying terms through applications (Dietterich 2002; Datar et al. 2002; Mansalis et al. 2018). The sliding window approach is employed in this study.

Streaming Data Clustering

Stream clustering on the other hand is a more recent, active area of study, as a result of the increased popularity of data

streams (Nguyen et al. 2015; Dasgupta et al. 2018). A data stream is a continuous flow of data where the system has no control over the volume of data that is arriving, and only a small fraction of the data is archived while the remainder is processed offline (Babcock et al. 2002; Dasgupta et al. 2018). Clustering in data streams is usually used to provide a real-time view on highly dynamic data. Some of the reasons why data streams differ from conventional data sources, in terms of data mining requirements, are (1) they are continuously flowing and never ending in nature, (2) there is no knowledge of the complete dataset as compared to conventional datasets, and (3) random access to data is not possible due to the single pass constraint (Dasgupta et al. 2018; Gama 2010; Hahsler et al. 2017). As a result of these differences, there have been a number of studies around clustering of events within data streams. Silva et al. (2013) surveyed existing stream clustering methods and categorised them as methods that perform object clustering and those that perform attribute clustering. Object clustering is the most common and takes the form of a two-step approach. The first step is data abstraction and the second is the clustering step. Silva et al. (2013) define attribute clustering as the case with the objective to find “groups of attributes that behave similarly through time, under the constraints assumed in a data stream scenario”. Silva’s work describes a taxonomy that classifies the existing clustering methods in terms of seven aspects. These aspects are data structure, window model, outlier detection mechanism, number of user-defined parameters, offline clustering algorithm, cluster shape, and type of clustering problem. Aggarwal (2013) describes clustering within data streams in the context of the four main classical categories, in alignment to Han et al. (2011), namely partitioning, density methods, probabilistic methods, clustering in high-dimensional streams, and discrete and categorical stream clustering methods, while touching briefly on work extended to other data domains such as categorical, text, and uncertain data. Mansalis et al. (2018) evaluated the performance of different algorithms for data stream clustering.

These notable surveys and evaluations discussed highlight different aspects about stream clustering; the work of Aggarwal (2013) is more relevant to this paper as it can be related to the spatiotemporal clustering approaches discussed in previous sections. Based on the reviews, the main clustering algorithms, classified according to the basic approaches discussed above, include partitioning methods—CluStream, ClusTree and STREAM; density-based methods—DenStream, D-Stream (grid based); and hierarchical methods—BIRCH. The main clustering methods applied are k-means, DBSCAN, and BIRCH. In data streaming terms, the method we apply is under the category of attribute clustering, which is less commonly employed compared to object clustering approaches.

Relevance of Related Work to Event Clustering

Having reviewed the state-of-the-art in all three relevant topics, related to establishing clustering patterns, the greatest challenge to clustering of lightning events in real-time is posed by the streaming nature of the data. Important aspects to consider, and issues that have been found to be challenging for data stream clustering include the ability of a clustering algorithm to adapt to the underlying changes that happen within a data stream due to the evolving nature of the data; the ability to handle limited time; and the ability to handle multidimensional data. Some clustering algorithms such as CluStream prescribe the number of clusters that can be detected and this is problematic for streaming data particularly highly dynamic data. Spatiotemporal clustering methods having been researched for much longer cover a lot of cases and data types of spatiotemporal events. The majority of the spatiotemporal clustering methods are either grid-based or look at a circular/cylindrical neighbourhood which does not necessarily explore all types of connections that occur between events.

Based on these reviews, in order to achieve the first objective of this paper, we apply a clustering method that has both aspects of hypothesis testing and the partitional approach as categorised by Shi and Pun-Cheng (2019). The advantages of the method employed in this study are as follows: firstly, this method is adaptive; the number and size of clusters are not determined beforehand. The method is not limited by the highly dynamic and evolving nature of the data as connections between events are confined to a sliding window. Secondly, this method improves the ability to handle multidimensional cases, where the dimensions are defined by space, time, and several attributes of lightning strikes (multiple attributes). The graph-based clustering approach supports as much clustering criteria in a single framework, in line with the fact that events may possess multiple descriptive attributes that need to be accounted for in a clustering algorithm. This is important since multiple attributes as well as derived characteristics from the attributes can be attached to events. This allows the use of various threshold measures and criterion from different algorithms in one graph. In most cases, clustering algorithms use single predefined measures but do not provide a framework for aggregating attributes to establish more complex relationships and other derived attributes. Graphs allow for different and complex relationships through path traversals thereby extending point-based clustering to path clustering and path discovery. For example, this is useful in establishing a lightning trail along high-risk dry forested areas where clustering in linear paths is required. Third and finally, most clustering is radius-based and focuses on finding clusters in a circular

or rectangular neighbourhood. However, graphs allow different clustering shapes to be established, for example shortest path linear clusters, circular clusters, and free form shapes.

Visual Analytics of Clustering Events

The sections above discuss the comprehensive studies in event clustering and pattern discovery, as well as the advantages of the clustering method used in this paper, in line with the first objective. In conclusion, the review of existing event clustering methods highlighted the usefulness of visualisation of clustering patterns, and ability to perform interactive analysis thereof based on domain knowledge. This aligns with the second objective of this paper and a state-of-the-art review of visualisation and visual analytics in the context of spatiotemporal events is thus discussed.

The visualisation and visual analytics of events, which forms the second objective of the paper, also remain an active field of study. Interesting research topics in visualisation and visual analytics of spatiotemporal events have emerged in literature over time, including those that focus on visual analytics at multiple levels of details (Silva et al. 2019), temporal evolution of events (Lukaszczuk et al. 2015), and frameworks (Andrienko and Andrienko 2016; Robinson et al. 2017). Since the importance of user input and inclusion of domain knowledge in pattern discovery and spatiotemporal analysis has been highlighted in recent studies, this can be fully exploited with the ability to visualise the results and make meaningful deductions. Andrienko et al. (2015) proposed an algorithm for real-time detection and tracking of spatio-temporal event clusters. Their work further developed a visual analytics system that consists of a dynamic map and an interactive interface that allows for changing clustering parameters and visualisation of results. This work is significant as it highlights two important concepts that are used as reference in this paper. Firstly, clustering of spatiotemporal events is more effective when allowing input of user domain-specific parameters interactively. Secondly, visualisation is an important step in clustering, as it allows the analysts to review their results which if not done properly can be difficult to deduce. Therefore, visual analytics is critical to understanding and reviewing patterns that result from the clustering. Visualisation of dynamic events is also affected by the characteristics of streaming data as discussed in Section “Streaming Data Clustering”. In the case of lightning data, the fact that the events occur in short burst and are irregular makes it particularly difficult to determine and visualise the areas affected in real-time. An option for visualisation and exploration of lightning events using the space time cube has been presented by Peters and Meng (2013). The basic approach of 3D visualisation in space and time is similar

to Peters and Meng (2013); however, we go a step further to demonstrate the visual effects of dynamic change over time (age of events and the patterns that emanate from clustering of lightning events over time). We also illustrate how interactivity within a 3D visualisation can be used in this case to overcome the challenges that result from this type of data in a real-time, streaming environment.

The method of determining patterns of lightning strikes in real-time presented in this paper provides two contributions to the existing knowledge as described in the reviews above. Firstly, in relation to the first objective of this paper, the clustering of spatiotemporal transient events, that occur unexpectedly with varying characteristics, combines hypothesis testing, a category described by Shi and Pun-Cheng (2019), and geometric graph-based principles (which includes aspects of partitional clustering), which have not been found to be applied much to this type of data in literature. The method also provides advantages based on challenges that were identified in literature as discussed in Section “[Relevance of Related Work to Event Clustering](#)”. Secondly, in relation to the second objective of the paper, the visual analytics approach makes use of the space time cube while adding interactive elements to visualise transient spatiotemporal events that occur unexpectedly and at irregular periods. We present in detail how the space time cube is formulated and constructed for the purpose of visualisation of dynamic, transient spatiotemporal events, while illustrating patterns that would otherwise remain unseen over time. While we make use of the space time cube as a visualisation tool, the scope of this study does not include a statement about the efficacy of the tool for the users. This would however be determined through conducting user studies, such as the framework developed by Kveladze et al. (2013).

Method

As a demonstration of a method for finding patterns in unexpected, transient events, clusters of lightning activity are determined in a short term thunderstorm. The data is made available through a data stream. The clusters of lightning are detected by a graph-based method and visualised in an interactive space time cube, to show progression of the event over time. The method for determining real time cluster patterns for lightning events follows a visual analytics approach, illustrated in Fig. 1, and detailed in the sections that follow.

Overview of Method

Following the visual analytics methodology illustrated in the framework, Fig. 1, the first step of the algorithm for lightning cluster detection in a real-time data stream, is the

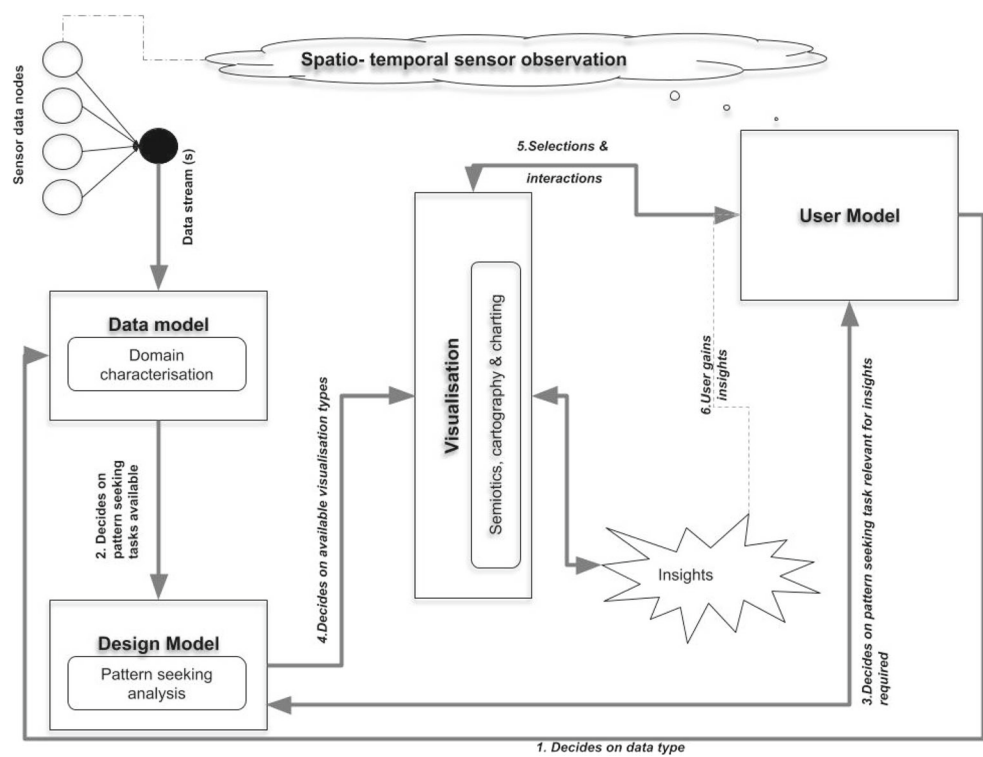
setting of the “User Model”. The user model contains all information that the user provides as inputs to the method. The five decisions made by the user model are as follows; firstly, selection of the area of interest (AOI), which at this point is defined by the rectangular bounding box of a user’s real-world area of interest. This area has a dual purpose; it sets out the area of analysis as well as the user’s view port of interest during visualisation of results. The second step is the setting of the observation period of interest. This is also used both during the data analysis process as well as by the visualisation model in setting up the view time of interest (VTol). Thirdly, selection of the data stream endpoint, in this case, a WebSocket endpoint that is streaming swordfish common data model (McFerren and van Zyl 2016) formatted data. The fourth step of the user model is selecting the pattern discovery method and setting the user-defined data filters and threshold values. In this case, lightning strikes of interest based on the attributes, and proximity thresholds are defined. The fifth step is selecting the visualisation tools and associated interactions to display the results. These data filters are used as inputs for the pattern discovery analysis.

The user model is marked up as an ECMA 404 JSON formatted document. This format can be interpreted and transferred easily within different parts of the framework represented in Fig. 1, shown as arrows in the diagram. This encoded document is sent as a settings document to both the design model and to the visualisation application. Parts of it are used to create a payload used by the stream processor within the design model.

The second step of this methodology is the determination of the “Data Model”. In agreement with the literature, in order to detect significant lightning clusters within a storm event, we require some domain knowledge. The understanding of the domain and characterisation thereof is represented by the “Data Model” in the framework diagram Fig. 1. Important information that is considered in this approach includes the type and characteristics of significant lightning strikes, the location and distance between incident lightning strikes, and the observed time of the strike. These properties are found in the decoded sensor observation message, which is derived from the data stream. The data used in this paper is described in Section “[Lightning Sensor Observations](#)”.

The third step takes place in the “Design Model”, where the user inputs as defined by the “User Model” and the data described by the “Data Model” are used for pattern-seeking analysis, which in this case is the determination of clustering patterns in spatiotemporal, irregular, and transient events—lightning strikes. The algorithm used for the detection of clusters in lightning events that take place within the design model is described in Section “[Clustering of Lightning Events](#)” and Section “[Algorithm Implementation](#)”.

Fig. 1 Conceptual model for determining real-time patterns from sensor observations (from Sibolla et al. (2018))



Following the detection of clusters of lightning activity, the cluster outputs are visualised according to step 4, that described the “Visualisation model” in this case using a space time cube where cluster formation, shape, age, and progression over time is illustrated based on semiotics and principles of cartography. This is described in Section “[Visualisation Method](#)”.

Lightning Sensor Observations

Following Fig. 1, the data model provides a description of the data and domain of lightning observations. This understanding of the domain is significant as it facilitates the interpretation of the patterns and visualisations discovered from the lightning events. Lightning activity can be classified into four kinds: cloud to ground (CG), cloud to cloud (CC), inter cloud (IC), and cloud to air. Cloud to ground lightning is generally associated with thunderstorms and destructive behaviour (Mofokeng et al. 2019). Therefore, in this study, we will be focusing on cloud to ground lightning strikes.

While the method described in this paper is not dependent on study area, it was tested with lightning activity within South Africa; hence, the clustering thresholds will be discussed in the South African context. South Africa is a lightning-prone country, which experiences significant lightning-related damage and loss of life (Mofokeng et al. 2019). The northern regions of the country exhibit a significant amount of lightning activity which is related to

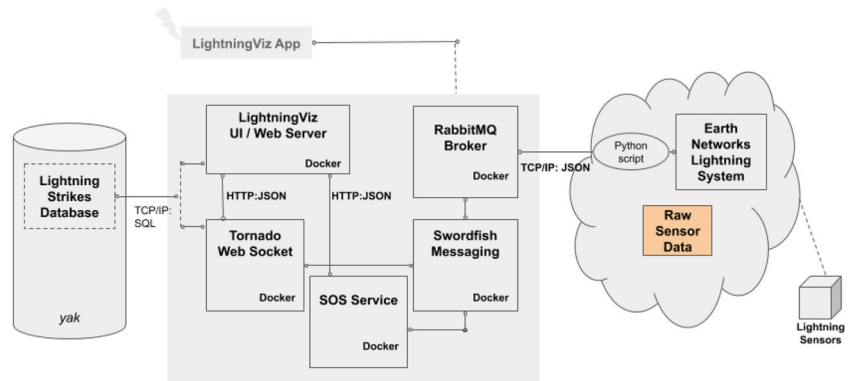
flash floods, short-term thunderstorms, and wildfires (Frost et al. 2018; Adepoju and Adelabu 2019).

The data used in this study is derived from the South African Advanced Fire Information System (AFIS) procured sensor network. These sensors are used by AFIS specifically to assist with the detection of fire ignition sources within well-known fire ecosystems across South Africa. The sensors report the location of a lightning strike as well as other properties of the strike. As illustrated in Fig. 2, these sensors report their observations in real time to a central server where they are then streamed in real time. The data is decoded, transformed into the swordfish common data model (McFerren and van Zyl 2016; Sibolla et al. 2018)—which is modelled from the OGC Observations and Measurements standard—and transported through an MQTT gateway. The implementation of MQTT is similar to the OGC Publish and Subscribe standard; however, at the time of this exercise, no other direct implementation of the standard could be found, and therefore MQTT was used. Standards compliance ensures that the method can be reused, following adherence to the same standard.

Clustering of Lightning Events

The clustering of lightning events references the design model in Fig. 1, which refers to determining patterns from the data. The clustering algorithm presented in this paper is based on the use of graphs in establishing

Fig. 2 A standards compliant architecture of the lightning sensor data; from sensor to the visualisation application



pairwise relationships and forming clusters. The graph-based algorithm consists of four major steps. Firstly, establishing pairwise relationships of lightning events using Delaunay triangulation; secondly, a graph of lightning observations is created from the triangulation followed by the retrieval of clusters using connected components. Finally, alpha shapes algorithm is used to retrieve concave cluster boundaries. The detailed description of each of these steps is described in the following section.

Determining Pairwise Relationships Between Observations

The clustering begins with the creation of a Delaunay triangulation, where the vertices are defined by the locations of the lightning strikes $e_{(x_i, y_i)}$. According to Lee and Schachter (1980), a Delaunay triangulation of a set of discrete points $V = \{v_1, \dots, v_n\}, N \geq 3$ in a Euclidean plane is such that the circle circumscribing any three points in the triangulation contains no point of the same set inside it. Lee and Schachter (1980) describe Delaunay triangulation extensively with the use of lemmas and corollaries and provide two algorithms for the implementation. The Delaunay triangulation presents a data structure that finds the closest optimal neighbour to each event thereby establishing pairwise relationship of only the next closest event, thus making it suitable in creating a graph. The pairwise relationships from the triangulation are used in the next section in creating the graph data structure. In other words, the triangulation is stored in a graph structure.

Graph Data Structure for Lightning Observations

By definition, graphs are designed to find connections and relations between discrete components based on any given attribute. Since clustering is based on establishing and finding relationships between objects, graph theory was thus employed in this study. According to Bondy et al. (1976), Wilson (1979), and Phillips et al. (2015), a graph is made up of vertices and edges as defined in Eq. 1. Vertices are then

connected by edges or links based on any chosen attribute. An edge links two vertices thereby establishing pairwise relationships. From the Delaunay triangulation, a graph is created using the triangulation vertices as nodes and the triangulation edges as graph edges. While adding the edges to the graph, attributes of each edge are calculated and added to each edge. The attributes used in this paper are Euclidean distance, type of lightning strike, and intensity.

$$G = (V, E)$$

where : G represents the graph
 V is a set of vertices
 E is a set of edges

(1)

In line with the principles of graph theory, in this study, the graph represents the network of spatiotemporal sensor observations, the sensor observations are equivalent to the vertices, and the edges define the relations between two observations. The graphs used to define the sensor observations are undirected and simple; the relationships between the observations are bi-directional (i.e. undirected edges) and they do not contain loops as defined in Fig. 3.

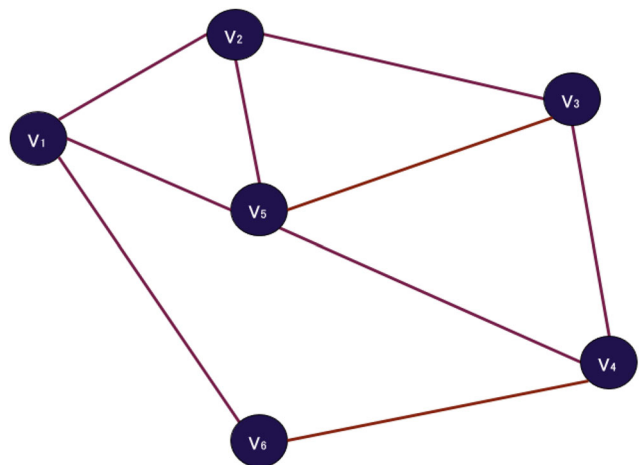


Fig. 3 Illustration of a simple undirected graph

Retrieval of Lightning Clusters

In order to find clusters of lightning events, connected component analysis is used. Connected components analysis (Cormen et al. 2009; He et al. 2017) is a graph theory algorithm that is used to establish a set of connected nodes or vertices, based on predefined criteria that are determined based on edge attributes, such as but not exclusive to distance. The diagrammatic representation presented in Fig. 4 shows how clusters are formed with a simplistic case that uses only distance as a criterion. In the figure illustration, edge weights are calculated based on the distance between two incident nodes, a threshold weight of 2 is used in the connected component analysis and clusters of points within a distance threshold of 2 are identified. This then leads to the identification of three components as seen on the figure on the right. In this case the connected components then represent the clusters. In real world application the threshold is often set as a result of domain knowledge. In addition to distance and attribute based criteria, the edge weights can be based on an unlimited set of attributes and/or equations that can be used to determine a relationship.

Retrieval of Cluster Boundaries

The output of the connected component analysis is a grouping of points that belong to the same cluster that meets the predefined threshold measure. The next step is to get the bounding polygon that encompasses the points in a component. In this step, based on the clustering of points discussed above, alpha shapes are used to define the clustering areas as polygons. A compact shape representation of a set of unorganised points in a cluster

is required to extract the boundary of the cluster. Since the boundary can either be concave or convex, a boundary retrieval method that can approximate both shapes optimally without area overestimation is required. Alpha shapes have proven to be the most efficient in reconstructing the best geometrical shapes from a point set. Convex hull-based methods are limiting since they tend to overestimate areas. Given a set of unorganised points, alpha shapes algorithms reconstruct the best shape that follows the point set boundary (Edelsbrunner 1992; Edelsbrunner and Mücke 1994; Bernardini and Bajaj 1997).

Alpha shapes are used to describe a generalisation of the bounding polygon that contains a set of points. Given a set of points P in a Euclidean plane as before, where:

$$P = P_i (i = 1, \dots, n) \text{ and } n \in R,$$

with the Delaunay triangulation of the points representing a graph:

$$G(V_p, E_p),$$

where V_p is a set of vertices and E_p is a set of edges. The alpha shape is described as the sub-graph:

$G_\alpha(V_\alpha, E_\alpha)$ where each edge is connected by any two points that lie on the boundary of a circle with a radius α (Mapurisa 2015). In this step, cluster boundaries are formed for visualisation.

Algorithm Implementation

Algorithm 1 shows the implementation details of the clustering method described above and is discussed in this section.

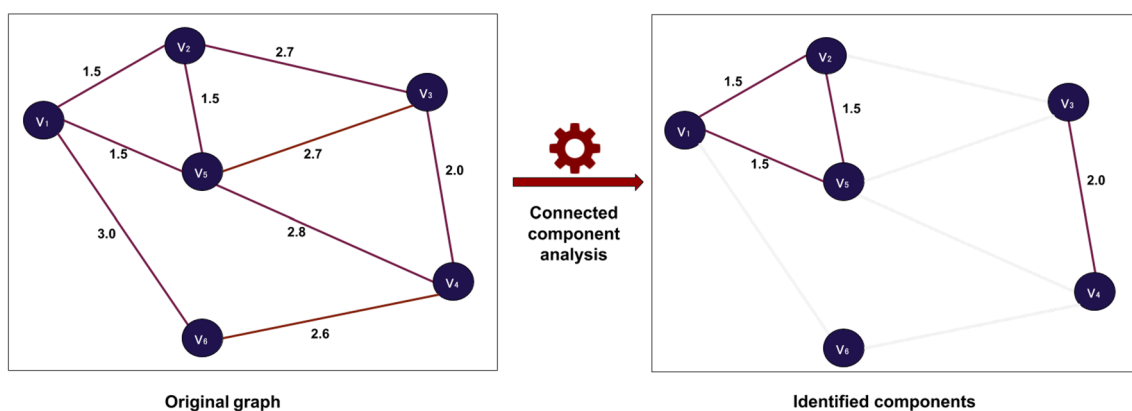


Fig. 4 An illustration of connected component analysis, showing the complete graph with edge weights on the left, and the identified connected components based on maximum edge weight of 2 on the right

Algorithm 1 Finding clusters in spatiotemporal sensor observations.

Input: Input JSON formatted user settings acquired from through a user interface

Input: Input JSON formatted messages from sensor observations in Swordfish Common Data Model

Output: JSON formatted lightning event cluster polygons detected by the clustering method applied

- 1 Read user inputs from the JSON document acquired from the user interface
- 2 Read JSON formatted sensor observation messages received from sensors
- 3 From incoming messages create, list T such that, there exists: $t_i \in T$

$$\text{where : } T = [t_s \leq t_i \leq t_e]$$

$$\text{and : } t_s = \text{start time}$$

$$t_e = \text{end time}$$

t_s and t_e are acquired from the JSON document read in (1)

- 4 Creating bounding box B such that, there exists: point $P = (x_j, y_j)$ inside B

$$\text{where : } x_j \in X = [x_{\min} \leq x_j \leq x_{\max}]$$

$$\text{And : } y_j \in Y = [y_{\min} \leq y_j \leq y_{\max}]$$

B is created from the user input AOI , acquired from (1)

- 5 For each time stamp within the interval selected follow step 6:

- 6 **for** ($t_i = 1$ to $\text{length}(T)$) {

1. Load all event messages e from the input data stream where;
 $e_i = \text{observations at time } t_i$
2. Create a graph structure using Delaunay triangulation for all;
 $e_{(x_i, y_i)} : (x_i, y_i)$
3. Assign weights to edges
4. Perform connected component analysis
5. Get clusters for t_i using alpha shapes

- 7 end for

- 8 **return** alpha shapes and cluster points JSON

As shown in the algorithm, the first step is to read the user inputs from the user settings document. This step sets the filtering requirements as defined by the user, namely *area of interest* and *viewer time of interest* and the threshold criteria. The viewer time of interest thus defines the total period of observation as well as the size of the time interval. Once the user-defined area of interest and viewer time of interest are set, the second step reads the incoming messages from the data stream. Steps 3 and 4 filter the incoming sensor messages to keep only those that are within the user-selected bounding box and time period. Clustering of lightning strikes is done per time interval as selected by the user, and within the area of interest of the viewer. In step 3, the incoming messages are filtered by input time such that messages that occur during the same time t_i are processed together, where t_i falls between the start and end times. Step 4 of the algorithm involves message filtering based on whether the data falls within the area of interest of the viewer. An observation falls within the area of interest if the coordinates of such a point lie within the user-defined bounding box. This defines observations

$e_{(x_i, y_i)}$ which are then processed to determine time-varying clustering of events in step 5.

Once all events that occur with a time step are loaded, the pairwise relationships are determined as described in Section “[Determining Pairwise Relationships Between Observations](#)” using the Python library “SciPy.Spatial” to construct the triangulations. A triangulation based on a selected area in South Africa, within one time stamp (5 min), is shown in Fig. 5.

The Delaunay triangulation above is used as an input for the connected component analysis as shown in Fig. 4. This is done using the SciPy Python library module `scipy.sparse.csgraph.connected_components`. A distance threshold is used in edge filtering leaving only the required connection between events within the graph. This threshold is set as a user variable—4 km used in this case. Previous studies of historical (retrospective) data have shown that clustering activity of lightning strikes happens within a range of 1–8 km within the study area (Adepoju and Adelabu 2019; Mofokeng et al. 2019).

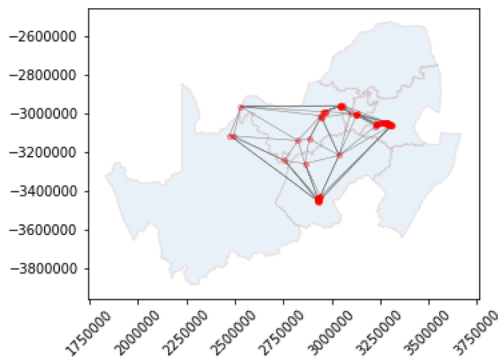


Fig. 5 Delaunay triangulation of lightning observations within a 5-min time step across South Africa

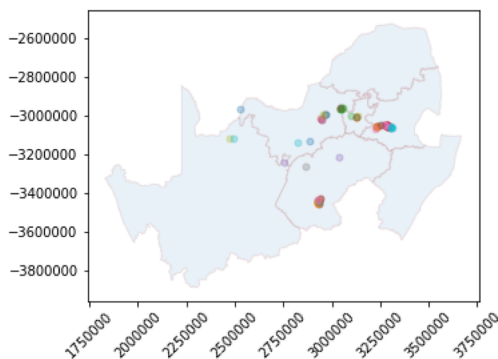


Fig. 6 Results of connected component analysis on the lightning data

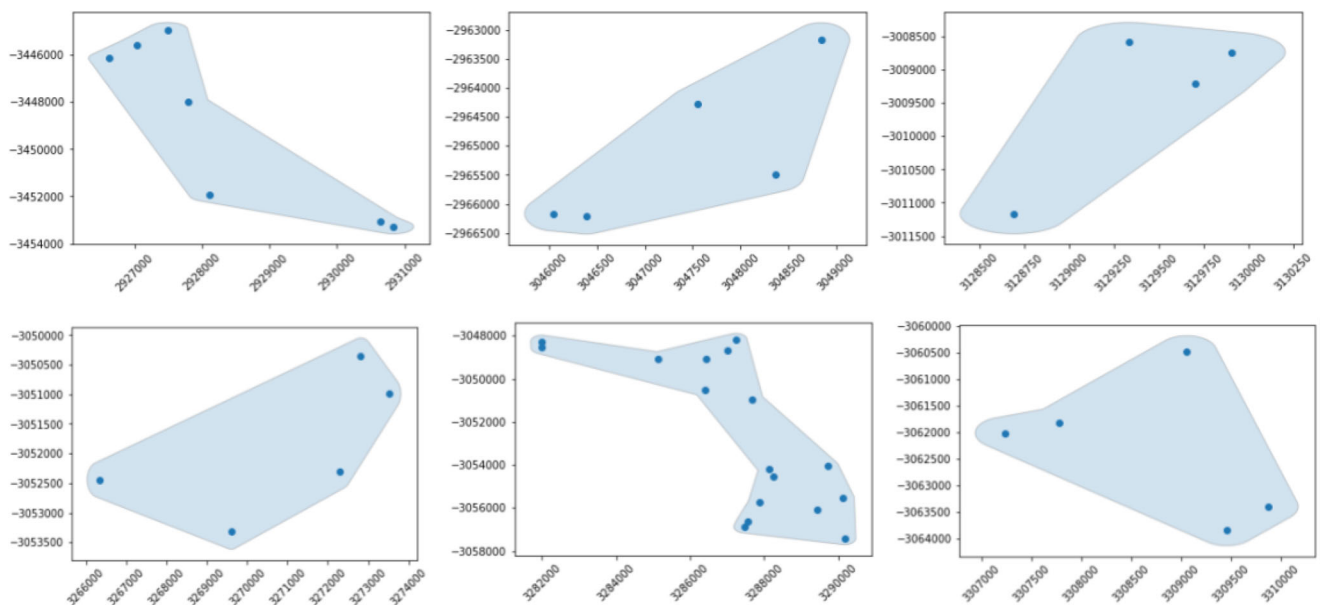


Fig. 7 Detected lightning cluster polygons based on the alpha shapes method

The connected component results from the above dataset are shown in Fig. 6. Points that belong to the same cluster have a similar color.

The output of the connected component analysis is a grouping of points that belong to the same cluster that meets the predefined threshold measure. The next step is to get the bounding polygon that encompasses the points in a component. In this step, based on the clustering of points identified above, alpha shapes are used to define the clustering areas as polygons.

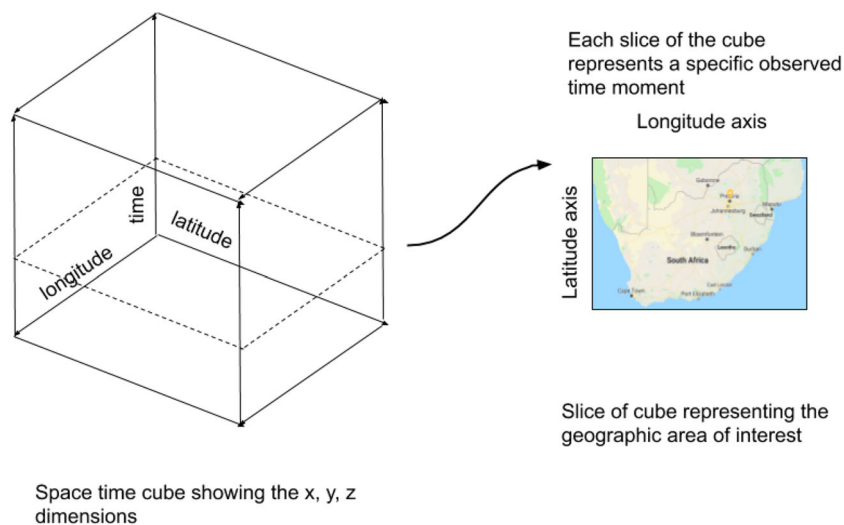
In this study, the alpha shapes were implemented using the Python alphashape 1.0.2 library. Following on from clusters identified above and depicted in Fig. 6, the cluster bounding polygons were determined as shown in Fig. 7. The alpha shapes were buffered by a distance of 1 km to illustrate the area of influence of the lightning strikes at the outer boundary.

Step 5 of algorithm 1 is processed iteratively for all time steps t_i until the end time set by the user is reached, and all clusters within this time range have been detected.

Visualisation and Interaction

Visualisation and interaction form the visualisation step of the methodology discussed in Fig. 1; for this application, the visualisation is done through a Web-based application that is developed by the authors, geoStreamViewer. The visualisation of transient observations is guided by (1) the viewer’s area of interest, (2) the observed properties

Fig. 8 Conceptual illustration of the space time cube



of the phenomenon being observed, and (3) the viewer's time of interest for observations. In this case, the lightning strikes had already been filtered in the clustering algorithm; therefore, all lightning strikes received were significant, and belonged to a cluster, and therefore no further categorisation was required. The cluster patterns that are derived represent the occurrence of significant lightning strikes during a user-selected time period. The visualisation of these patterns aims to display occurrence by highlighting birth and cessation of a clustering transient event. The age of the observed patterns, between birth and cessation is displayed to the user. A Space Time Cube, created with CesiumJS, with a timeline and clock, is used as the visualisation environment for the resulting data.

The space time cube was developed as early as the 1970s (Gatalisky et al. 2004; Kraak 2003) to represent two-dimensional space and adding time as a third dimension to illustrate spatio-temporal movements and events. Several adaptations of the space time cube have been developed to visually show patterns in spatio-temporal data. These adaptations of the space time cube include representation of spatial events (Gatalisky et al. 2004), temporal data (Bach et al. 2017), movement analysis (Andrienko et al. 2014; Demšar and Virrantaus 2010), and eye movement data (Li et al. 2010).

In this study, the space time cube is defined as a rectangular prism, where length and width are defined by the bounding box that encompasses a specific geographic area of interest; and the height of the rectangular prism represents the time stamps of the view time of interest, which can also be referred to as the time window of interest of the observations. Each slice of the rectangular prism represents a single time moment, which is equal to the observation time, as in Fig. 8. The surface area of each slice represents the thematic distribution of a selected observed property that can be modelled to display

the patterns discussed, such as heatmaps, choropleths, or discrete classified points.

Visualisation Method

Algorithm 2 Visualisation of lightning clusters.

Input: Input JSON formatted user settings

Input: Input JSON formatted clusters description

Output: Space time cube visualisation lightning event cluster polygons

- 1 Set up base of space time cube using the user's AOI description
- 2 Set up visualisation timeline using the user's Viewer Time of Interest (VToI) description
- 3 Set up access to the WebSocket to receive processed cluster messages
- 4 Get description of the visualisation method from user configuration
- 5 Create CZML data visualisation object
- 6 Update czml with new clusters as they become available Display time oriented clusters on space time cube

The first step in the visualisation process is to read the user configuration from the JSON input. How these user configuration variables are used are described in the following 4 points:

1. The area of interest described similarly to the clustering pattern algorithm above is used to define the base of the Space Time Cube. Instead of rendering the whole globe in CesiumJS, only the area of interest is displayed to enhance in-browser performance, and to help the user focus on their data region of interest excluding unnecessary data.
2. Viewer Time of Interest is defined by the start time, end time, and time interval (in seconds, minutes, or

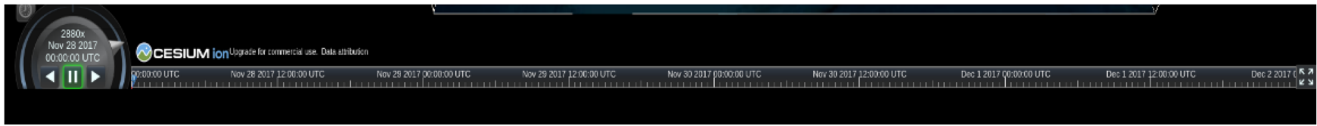


Fig. 9 CesiumJS, Animation, and timeline widgets

hours). These user properties are used to configure the timeline and the clock widgets attached to the Space Time Cube visualisation. The start and end times are used to define the start`Time` and stop`Time` of the timeline widget as defined in `CesiumJs`. These times are provided using the ISO8601 time standard, with assistance of an added date time picker widget. The interval within the visualisation model is used to define the animation time steps (`clockStep` in `CesiumJs`) on the timeline (Fig. 9).

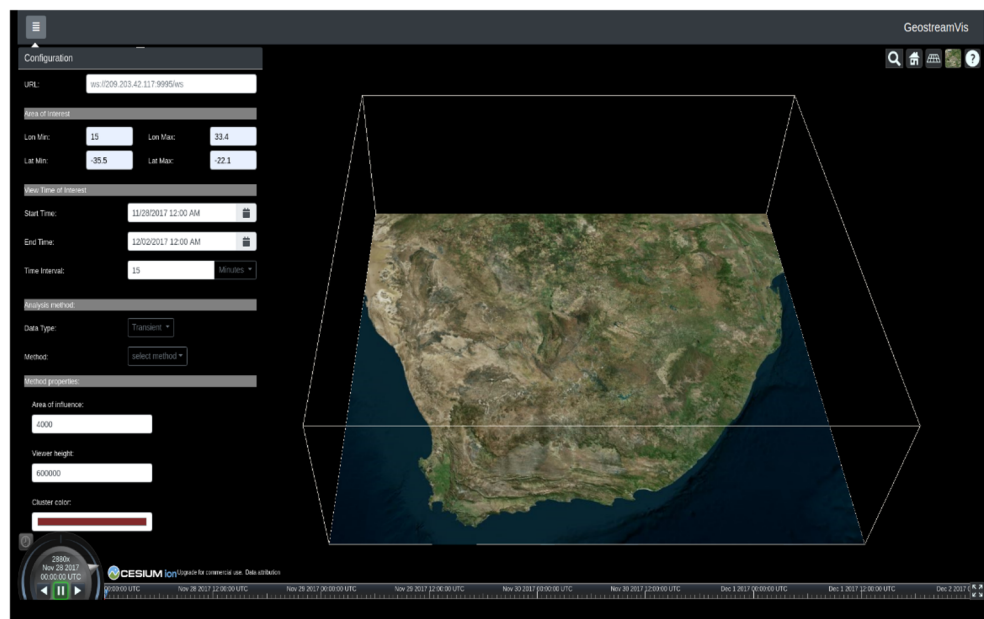
3. The URL for the data source is provided in the form of a `WebSocket`; for real-time data, this method provides the data in `JSON` format.
4. A description of the type of observations to be visualised. This allows for the filtering and matching of pattern methods to data types to guide the user about relevant methods for their data.

The visualisation of clusters in `geoStreamViewer` requires the user to provide the area of influence as mentioned in the cluster pattern algorithm discussed above. The area of influence is then used to define a buffer distance, once the bounding shapes are determined. The user is also allowed to select the base colour used to present the cluster polygons, as well as the preferred height of the space time cube. An illustration of how these are applied follows.

Having configured the visualisation environment as per the user needs, the following step is the time-based visualisation and animation of the cluster polygons. The steps towards this are:

Setting up the Space Time cube as a `CesiumJS` viewer. The viewer is set up such that it allows for rendering of the cube in 2D and 3D perspectives, with a choice of either satellite image or street map background images, which also define the base of the cube. This is shown in the screenshot of the viewer in Fig. 10. Once the space time cube has been set up, the following step is to receive the `JSON` objects defining the derived cluster polygons. The results of the pattern discovery method described in Section “Algorithm Implementation” are received as a stream of time stamped `JSON` formatted objects that describe the cluster polygons and points within them, the time interval in which the clusters formed, and any additional properties that describe the clusters. Although `CesiumJS` is able to visualise the `GeoJSON` data format, the native `CesiumJS` object for visualisation of animations, `Cesium Markup Language (CZML)` is used as it was found to be more suitable for streaming data. The overall structure of `CZML` is described in Zhu et al. (2018). A `CZML` document is `JSON` based and consists of a list of `JSON` formatted objects stored in an array, called packets. The first packet consists of a description of the document; the rest of the packets describe the data to be visualised in the viewer.

Fig. 10 CesiumJs, Animation, and timeline widgets



For every time stamped data object that is received from the stream, a CZML packet is created and the CZML document is updated. The definition of the packets is such that it highlights the birth, duration, and cessation of an event, as well as to describe the behaviour of this event during its period of existence. The availability property of the packet is used to ensure that the event is tracked throughout the users' view time of interest as discussed in previous sections. The event is defined by the polygon property as a time-varying object. The polygon property includes the positions of the event. The positions of the event describe the geometric coordinates and the interval during which these positions are valid as well as the description of how they are displayed. Each geometric description is accompanied by the interval and style that is used to display it during this interval. The custom style object developed for this visualisation is defined by:

$$\begin{aligned} \text{style} &= f((rgb), \alpha, t_i) \\ &= [\text{color.r}, \text{color.g}, \text{color.b}, \alpha] \\ \text{where : } &rgb = \text{color} \\ \alpha &= ((t_i - t_s)/(t_c - t_s)) \times 255 \\ \text{and : } &t_s = \text{viewer's start time} \\ &t_c = \text{current viewer time} \\ &t_i = \text{event observation time} \end{aligned} \quad (2)$$

The style object is designed such that one colour is used to represent the event, and the colour fades out with the age of the event, to the point that the event polygon tracks are translucent when the event has ended (ceases to exist). The colour used to represent the event is as per the viewer's preference. Since the data is binary, in the sense that only occurrence or non-occurrence of the event is monitored, it would not add any value to use multiple colours to visualise the polygons.

The height variable in the space time cube is also meant to show the progressive nature of the events that occur using the z axis (height attribute); with the thickness of the height slice representing the duration of an event. The height of each time stamped event is defined by:

$$\begin{aligned} h_e &= f(t_c, h_c) \\ &= (t_i/(t_e - t_s)) \times h_c \\ \text{where : } &h_e = \text{cube height of event} \\ &h_c = \text{total cube height} \\ &t_e = \text{viewer's end time} \end{aligned} \quad (3)$$

The height of each event cluster slice is defined by:

$$\begin{aligned} \text{extrusionheight} &= \text{heightslice} \\ \text{heightslice} &= h_s \end{aligned} \quad (4)$$

Once a packet is successfully created, it is added to the main CZML document and the visualisation is updated. This

procedure is repeated for all time-stamped clusters that are detected within the viewer-specified period.

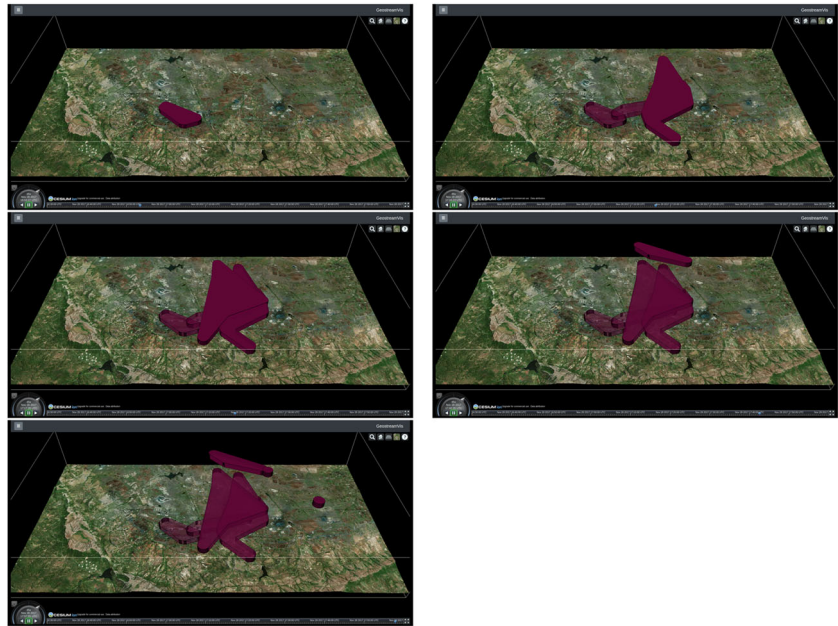
Visualisation Results

The outcome of the time varying visualisation process is shown in Fig. 11. The figure shows 5 time steps of lightning events that were clustered per 5-min interval. Time "T1" records the birth of the event, when the first lightning incident occurred. At "T2", the region affected by the lightning strikes starts increasing. As the lightning strikes and cluster regions grow over time, the clusters that formed earlier and during birth of the event begin to show age and slowly become faint in colour. At "T5" (the final diagram), the event has ended, but since it occurred during the viewer's time of interest the footprints of the occurrence remain, and if the user were to scroll back in time along the timeline widget they would still be able to see how the event unfolded as it happened. This illustration is relevant for a short-term visualisation of a single event made up of multiple time-varying clusters of incidences.

Figure 12 provides a wider picture of a longer term view time of interest. In this figure, the viewer's period of interest is increased to 1 week, over the whole of South Africa. The same principle of fading colour over time is still applied. In addition to there being more data in the cube, a new trend is observed. The data when extruded along the full length of the cube shows darker shaded and lighter shaded columns of different heights. The shorter columns represent events that occurred over a short time within the week-long period, whereas the taller columns occurred over a longer period. The lighter shaded columns show events that occurred earlier on in the week, closer to the start time of the user's week-long period. The darker shaded periods show more recent events within the viewer's time of interest. There is also a variation in thickness of the columns. The thicker columns represent events that covered a wider region as opposed to thinner columns that show events that covered a narrower region. These explanations can be provided to the user in the form of a map legend.

The space time cube can also be flattened from three dimensions into a two-dimensional map; this view when used with the timeline animation allows the user to view the time-based progression of an event with a closer view on how the event behaves through time. This is a view that is most common for visualisation of animations. This feature is important because it guides users who are not familiar with the space time cube and data visualisation in 3 dimensions to slowly ease into it while still being able to revert to a more familiar option.

Fig. 11 Progression of a lightning storm event, illustrating the age of observations by (z attribute) and colour on a Space Time Cube. The images refer to time steps “T1” to “T5” progressively



Discussion

This paper demonstrates a methodology for determining real-time patterns in lightning strikes from sensor observations, where lightning strikes represent irregular transient observations that are characterised by highly dynamic and variable attribute values. A methodology for determining clustering patterns in these types of events is illustrated with the intention of showing areas that may be adversely affected by the occurrence of these events timeously, such that mitigation or protection strategies can be employed.

The methodology developed in this paper follows the visual analytics approach that includes, firstly, spatiotemporal analysis for the detection of patterns; secondly, allowing human interaction to provide domain expertise in order to

guide the analysis process; and thirdly, visualisation to show the resulting patterns in real time using concepts of cartography and semiotics, to illustrate the behaviour of the phenomenon of study as it unfolds.

The use of algorithms within the paper aims to provide not only spatiotemporal analysis methods and visualisation but also to provide technical guidelines for application developers undertaking similar work.

The contribution of this paper is based on two objectives, firstly to show how graph theory can be used to determine clustering patterns for irregular occurring transient observations of highly dynamic characteristics. The second objective is to illustrate how a space time cube can be used to visualise arising clustering pattern evolution in real-time.

Fig. 12 Lightning cluster events for a week across South Africa



We therefore discuss the results, advantages, and shortcomings discovered with the method in relation to these two objectives.

Transient observations, by definition, do not occur at regular times; therefore, algorithms for discovery of patterns and visualisation need to be highly adaptive and dynamic since observations can either be very dense or sparse.

The use of a graph-based method for detection of clustering patterns of lightning strikes in real-time proved to have advantages for these types of data. Firstly, the method proved to be adaptive to the underlying characteristics of the data. Segmenting the data into sliding temporal windows allowed the stream of lightning sensor observations to be analysed as it evolved; hence, concept drift was not an issue, as each analysis window is self-contained. As with some other methods, the number of clusters is not defined beforehand, and the graph connections determined the number and size of the clusters driven by the amount of data received. The second advantage of the use of graphs is the ability to allow multiple types of clustering criteria. Sensor observations of lightning strikes have multiple dimensions defined by the space, time, and several descriptive attributes. In this study, the clustering criteria used were the Delaunay triangulation to determine the distance between observation and time interval between lightning strikes to determine the temporal sliding window for analysis, as well as attributes of the observations, namely type of lightning strike and intensity. The third and final observed advantage of this clustering method is that it allowed for the formation of free-form shape of lightning cluster polygons. The connections between lightning event points were not based on a regular-shaped region, therefore allowing for more connections between events to be detected. The use of alpha shapes to delineate the cluster boundary shapes as opposed to a convex hull prevented the overestimation of affected area also resulting in more realistic shapes.

Visualisation of transient irregular events faces the challenge of overcrowding and sparsity in different regions of the same area of interest. This challenge has been overcome by allowing user-configured parameters namely height and start and end times of the timeline, for the space time cube viewer. It has been found that setting shorter time periods of view enables the data to be distributed “freely” within the cube. Alternatively, if longer time periods of visualisation are desired, similar to Fig. 12, then user interaction becomes a key factor. One of the major advantages of choosing CesiumJS for development of the Space Time Cube is that it has been possible to allow the user to zoom in and out and fly between visualised items in order to get a closer look and visually explore the data more intuitively.

If the data is too dense or sparse, a further enhancement to the visualisations that could be implemented is the binning

of the visualisations. In this case, the data was binned and limited to the height of the space time cube so as to ensure that it does not stack to an uncontrollable height which would make it less intuitive. A related issue occurs when the time period is too long: the change in colour, which represents the age of the observations, becomes either too gradual or too abrupt depending on how fast and frequent the data is arriving. During this study, we found that implementing the style equation as shown in Eq. 2 allows for a gradual change in colour that is also relative to the stage of occurrence within the time of interest.

While the results of the clustering and visualisation of observations of lightning strikes in real-time seemed satisfactory, the usability of the visualisations for users was not assessed. Literature has shown that frameworks for user studies have been developed. It would be important to use one of the existing frameworks to assess the usability of the space time cube under such highly dynamic conditions where the data characteristics are rapidly changing, and the flow of data is highly variable.

Data Availability In line with the principle of code sharing, the authors have made the code developed for this paper available in the following repository:

<https://github.com/.../lightning-cluster-detection.git>

Compliance with Ethical Standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical Approval No humans or animals were involved; therefore, no ethical approval was required. Data and software were used according to terms and conditions in the respective licenses.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Adepoju K, Adelabu S (2019) Assessment of fuel and wind drivers of fire risk in protected mountainous grassland of south africa. In: IGARSS 2019-2019 IEEE International geoscience and remote sensing symposium, pp. 867–870. IEEE
- Aggarwal CC (2013) A survey of stream clustering algorithms
- Andrienko G, Andrienko N (2016) Gaining knowledge from georeferenced social media data with visual analytics. European Handbook of Crowdsourced Geographic Information, 157

- Andrienko G, Andrienko N, Demsar U, Dransch D, Dykes J, Fabrikant SI, Jern M, Kraak MJ, Schumann H, Tominski C (2010) Space, time and visual analytics. *Int J Geogr Inf Sci* 24(10):1577–1600
- Andrienko G, Andrienko N, Schumann H, Tominski C (2014) Visualization of trajectory attributes in space–time cube and trajectory wall. In: *Cartography from pole to pole*. Springer, Berlin, pp 157–163
- Andrienko N, Andrienko G, Fuchs G, Rinzivillo S, Betz HD (2015) Detection, tracking, and visualization of spatial event clusters for real time monitoring. In: *2015 IEEE International conference on data science and advanced analytics (DSAA)*, pp 1–10. IEEE
- Ansari MY, Ahmad A, Khan SS, Bhushan G et al (2019) Spatiotemporal clustering: a review. *Artif Intell Rev*, 1–43
- Babcock B, Babu S, Datar M, Motwani R, Widom J (2002) Models and issues in data stream systems. In: *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 1–16
- Bach B, Dragicevic P, Archambault D, Hurter C, Carpendale S (2017) A descriptive framework for temporal data visualizations based on generalized space-time cubes. In: *Computer graphics forum*, vol 36. Wiley Online Library, pp 36–61
- Bernardini F, Bajaj CL (1997) Sampling and reconstructing manifolds using alpha-shapes. Department of Computer Science Technical Reports Paper, 1350
- Bondy JA, Murty USR et al (1976) *Graph Theory with Applications*, vol 290. Macmillan, London
- Cormen TH, Leiserson CE, Rivest RL, Stein C (2009) *Introduction to Algorithms*. MIT Press, Cambridge
- Dasgupta A, Arendt DL, Franklin LR, Wong PC, Cook KA (2018) Human factors in streaming data analysis: Challenges and opportunities for information visualization. In: *Computer graphics forum*, vol 37. Wiley Online Library, pp 254–272
- Datar M, Gionis A, Indyk P, Motwani R (2002) Maintaining stream statistics over sliding windows. *SIAM J Comput* 31(6):1794–1813
- Demšar U, Verrantaus K (2010) Space–time density of trajectories: exploring spatio-temporal patterns in movement data. *Int J Geogr Inf Sci* 24(10):1527–1542
- Dieterich TG (2002) Machine learning for sequential data: a review. In: *Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition (SSPR)*, pp 15–30. Springer
- Edelsbrunner H (1992) Weighted alpha shapes. University of Illinois at Urbana-Champaign
- Edelsbrunner H, Mücke EP (1994) Three-dimensional alpha shapes. *ACM Trans Graphics (TOG)* 13(1):43–72
- Frost P, Kleyn L, van den Dool R, Burgess M, Vhengani L, Steenkamp K, Wessels K (2018) The elandskraal fire, knysna. CSIR Report number: 271960-1
- Gama J (2010) *Knowledge Discovery from Data Streams*. CRC Press, Boca Raton
- Gatalsky P, Andrienko N, Andrienko G (2004) Interactive analysis of event data using space-time cube. In: *Proceedings. Eighth international conference on information visualisation, 2004. IV 2004*, pp 145–152. IEEE
- Hahsler M, Bolanos M, Forrest J et al (2017) Introduction to stream: an extensible framework for data stream clustering research with r. *J Stat Softw* 76(14):1–50
- Han J, Pei J, Kamber M (2011) *Data Mining: Concepts and Techniques*. Elsevier, Amsterdam
- He L, Ren X, Gao Q, Zhao X, Yao B, Chao Y (2017) The connected-component labeling problem: a review of state-of-the-art algorithms. *Pattern Recogn* 70:25–43
- Iordanidou V, Koutroulis A, Tsanis I (2016) Investigating the relationship of lightning activity and rainfall: a case study for crete island. *Atmos Res* 172:16–27
- Keim DA, Mansmann F, Schneidewind J, Thomas J, Ziegler H (2008) Visual analytics: Scope and challenges. In: *Visual data mining*, pp 76–90. Springer
- Kisilevich S, Mansmann F, Nanni M, Rinzivillo S, Clustering ST (2010) *Data mining and knowledge discovery handbook*
- Kohn M, Galanti E, Price C, Lagouvardos K, Kotroni V (2011) Nowcasting thunderstorms in the mediterranean region using lightning data. *Atmos Res* 100(4):489–502
- Kraak MJ (2003) The space-time cube revisited from a geovisualization perspective. In: *Proceedings 21st international cartographic conference*, pp 1988–1996. Citeseer
- Kveladze I, Kraak MJ, van Elzakker CP (2013) A methodological framework for researching the usability of the space-time cube. *Cartogr J* 50(3):201–210
- Lee DT, Schachter BJ (1980) Two algorithms for constructing a delaunay triangulation. *Int J Comput Info Sci* 9(3):219–242
- Li X, Çöltekin A, Kraak MJ (2010) Visual exploration of eye movement data using the space-time-cube. In: *International conference on geographic information science*, pp 295–309. Springer
- Lukaszczuk J, Maciejewski R, Garth C, Hagen H (2015) Understanding hotspots: a topological visual analytics approach. In: *Proceedings of the 23rd SIGSPATIAL international conference on advances in geographic information systems*, pp. 1–10
- Mansalis S, Ntoutsis E, Pelekis N, Theodoridis Y (2018) An evaluation of data stream clustering algorithms. *Stat Analis Data Min Data Sci J* 11(4):167–187
- Mapurisa W (2015) Alpha shapes based point cloud filtering, segmentation and feature detection. In: *Africageo conference*
- McFerrer G, van Zyl T (2016) Geospatial data stream processing in python using foss4g components. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 41
- Mofokeng DO, Adelabu AS, Adepoju K, Adam E (2019) Spatio-temporal analysis of lightning distribution in golden gate highlands national park (gghnp) using geospatial technology. In: *IGARSS 2019-2019 IEEE International geoscience and remote sensing symposium*, pp 9898–9901. IEEE
- Nguyen HL, Woon YK, Ng WK (2015) A survey on data stream clustering and classification. *Knowl Inf Sys* 45(3):535–569
- Peters S, Meng L (2013) Visual analysis for nowcasting of multidimensional lightning data. *ISPRS Int J GeoInf* 2(3):817–836
- Phillips JD, Schwanghart W, Heckmann T (2015) Graph theory in the geosciences. *Earth Sci Rev* 143:147–160
- Podur J, Martell DL, Csillag F (2003) Spatial patterns of lightning-caused forest fires in ontario, 1976–1998. *Ecol Modell* 164(1):1–20
- Robinson AC, Peuquet DJ, Pezanowski S, Hardisty FA, Swedberg B (2017) Design and evaluation of a geovisual analytics system for uncovering patterns in spatio-temporal event data. *Cartogr Geogr Inf Sci* 44(3):216–228
- Shi Z, Pun-Cheng LS (2019) Spatiotemporal data clustering: a survey of methods. *ISPRS Int J GeoInf* 8(3):112
- Sibolla BH, Coetzee S, Van Zyl TL (2018) A framework for visual analytics of spatio-temporal sensor observations from data streams. *ISPRS Int J GeoInf* 7(12):475
- Silva JA, Faria ER, Barros RC, Hruschka ER, Carvalho ACD, Gama J (2013) Data stream clustering: a survey. *ACM Comput Surv (CSUR)* 46(1):1–31
- Silva RA, Pires JM, Datia N, Santos MY, Martins B, Birra F (2019) Visual analytics for spatiotemporal events. *Multimed Tools Appl* 78(23):32,805–32,847
- Strauss C, Rosa MB, Stephany S (2013) Spatio-temporal clustering and density estimation of lightning data for the tracking of convective events. *Atmos Res* 134:87–99

- Tork HF (2012) Spatio-temporal clustering methods classification. In: Doctoral symposium on informatics engineering, vol 1. Faculdade de Engenharia da Universidade do Porto Porto, Portugal, pp 199–209
- Wilson RJ (1979) Introduction to graph theory. Pearson Education India
- Wu F, Cui X, Zhang DL, Qiao L (2017) The relationship of lightning activity and short-duration rainfall events during warm seasons over the beijing metropolitan region. *Atmos Res* 195:31–43
- Zhu L, Wang Z, Li Z (2018) Representing time-dynamic geospatial objects on virtual globes using czml—part i: overview and key issues. *ISPRS Int J Geoinf* 7(3):97

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.