Experience: Quality Assessment and Improvement on a Forest Fire Dataset

ROGÉRIO LUÍS C. COSTA, IEETA, University of Aveiro, Portugal
ENRICO MIRANDA, IEETA, University of Aveiro, Portugal
PAULO DIAS, DETI - IEETA, University of Aveiro, Portugal
JOSÉ MOREIRA, DETI - IEETA, University of Aveiro, Portugal

Spatio-temporal data can be used to study and simulate the movement and behavior of objects and natural phenomena. However, the use of real-world data raises several challenges related to its acquisition, representation and quality. This paper presents a data cleaning process based on consistency rules and checks, that uses geometric operations to detect and remove outliers or inaccurate data in a spatio-temporal series. The proposal consists of selecting key frames and applying the process iteratively until the data have the desired quality. The case study consists of extracting and cleaning spatio-temporal data from a video tracking the propagation of a controlled fire captured using drones. The source data was generated using segmentation techniques to obtain the regions representing the burnt area across time. The main issues concern noisy data (e.g., the height of flames is highly variable) and occlusion due to smoke. The results show that the quality assessment and improvement method proposed in this work can identify and remove inconsistencies from a dataset of more than 22,500 polygons in just a few iterations. The quality of the corrected dataset is verified using metrics and graph analysis.

CCS Concepts: • **Information systems → Temporal data**; **Inconsistent data**; **Spatial-temporal systems**; **Data cleaning**.

Additional Key Words and Phrases: spatio-temporal data, data quality, data consistency

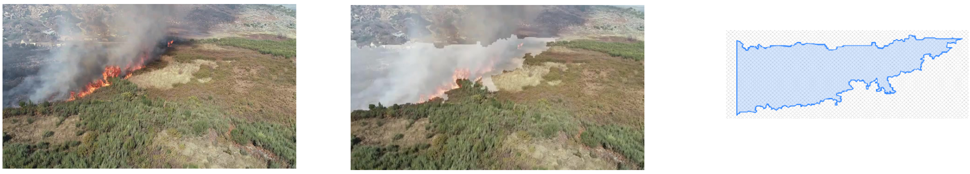
ACM Reference Format:
Rogério Luís C. Costa, Enrico Miranda, Paulo Dias, and José Moreira. 2020. Experience: Quality Assessment and Improvement on a Forest Fire Dataset. *ACM J. Data Inform. Quality* 1, 1 (July 2020), 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Spatio-temporal data allow the representation of the movement and behavior of objects and natural phenomena over time. Data models and query languages already exist to represent and analyze the evolution of spatial data over time, but most of research on modelling and querying spatio-temporal data uses synthetic datasets and there is little work focusing on transforming real-world data into spatio-temporal data compatible with existing data models. Also, the transformation of raw data into spatio-temporal data raises several issues since real-world data are often noisy, incomplete or invalid.

Authors' addresses: Rogério Luís C. Costa, rogeriocosta@ua.pt, IEETA, University of Aveiro, Portugal, 3810-193; Enrico Miranda, enrico.miranda@ua.pt, IEETA, University of Aveiro, Portugal, 3810-193; Paulo Dias, paulo.dias@ua.pt, DETI - IEETA, University of Aveiro, Portugal, 3810-193; José Moreira, jose.moreira@ua.pt, DETI - IEETA, University of Aveiro, Portugal, 3810-193.

1936-1955/2020/7-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn



(a) Video frame

(b) Segmented burnt area

(c) Burnt area representation at database level

Fig. 1. Burnt area representation: from video frames to database representation

We are particularly interested on using *moving regions* [12] to represent real-world phenomena that change continuously over time. This allows the execution of operations on the spatio-temporal behaviour of the phenomena, including the estimation of its evolution in the interval between observations. Our research includes the use of moving regions to represent the propagation of forest fires with application to studies on the emission of pollutants to the atmosphere.

The dataset used this work was generated from video sequences of controlled fires acquired using RGB cameras mounted on drones. The videos have been processed using segmentation techniques to extract the shape of the burned areas in each frame. The result is a dataset composed of thousands of shapes (i.e. polygons) with complex geometries, each one representing the burned area at a certain timestamp. These geometries were used together with region interpolation methods ([10, 20, 23]) to create the moving region representation of the burned area at database system level. Figure 1 shows an example of a video frame, the burned area obtained using a segmentation algorithm and its representation at database level (i.e. a polygon stored in a column of the geometry datatype in PostgreSQL).

The segmentation of real world videos on forest fires may comprise several challenges due to the number of heterogeneous objects such as lakes, different types of vegetation, roads, fences and burned area. Also, the erratic behaviour of flames and smoke may hide information from areas behind them and generate variable occlusion. The main problem here is to find data that does not reflect the real state of the modeled entity (i.e. burnt area) at the corresponding timestamp. Accuracy is the data quality dimension we aim to improve and inaccurate data should be removed from the dataset, as new representations may be generated (if necessary) at database level by applying region interpolation functions over accurate data.

If one considers each geometry individually, the only way to identify inaccurate representations is by visual inspection. But if one considers that each geometry is part of a temporal series, then it is possible to check for consistency problems. For instance, a tree that is inside the burned area at a given timestamp, should also be inside the burned area at all subsequent timestamps. This means that it is possible to use domain knowledge and specify rules to identify incorrect representations (e.g., outliers) in a time series where the values are polygons representing the state of a phenomenon at a given time. There are several models to detect outliers, but the wide of variations in problem formulations makes that it is often not possible to use off-the-shelf models in such context [13]. Also, the existence of clusters of outliers (i.e. clusters of inaccurate data in which polygons are consistent between them) can cause further challenges to outlier detection due to masking and swamping [8].

In this work, we use specialized methods (based on geometric operations and metrics) as part of a more traditional approach (with quality assessment and improvement). After defining quality rules, requirements and checking mechanisms, we start the quality assessment and outliers removal procedure. We define time ranges and manually verify (and adjust) the polygons corresponding to the first and last timestamps in the range. Then, we apply quality verification mechanisms to identify and remove inaccurate data in the range. Quality metrics are evaluated and, if necessary, new time ranges are defined and the quality assessment and outliers removal process is restarted for the new ranges. The rules and methods we present can be extended and adapted to process other real-world phenomena, like the glaciers retreating or the spreading of lava flows.

The following section presents an overview of the related work. Section 3 describes the dataset, as well as data acquisition, preparation and quality issues. Section 4 presents quality rules, requirements and checking mechanisms. Section 5 presents the experimental results. Section 6 concludes the paper and presents the future work.

2 RELATED WORK

In the database systems domain, outliers generally refer to data (resulting from anomalies, noise or unusual events) that are inconsistent with the rest of a dataset [7]. Cheng and Li [7] define a spatio-temporal outlier in terms of the difference between value in a spatio-temporal neighbourhood. They can also be viewed as erroneous representations of real-world objects or phenomena that possibly affect data quality [8].

Most of existing approaches can be classified into *distribution-based*, *depth-based* and *distance-based* [7]. In the context of temporal data, there are several works on outlier detection in time series ([4, 15, 19]) and in stream data ([2, 3]). Some examples of commonly used strategies are autoregressive models ([4, 14, 24]) and clustering ([1, 5, 11, 16]). But several factors impact on selection of detection strategy to be used, including the nature of the data and the relationship among data instances, and due to the wide variation of problem formulations, it is often not possible to use off-the-shelf outlier detection models in new contexts [13].

In [7], authors propose a 4 steps method to detect outliers in spatio-temporal data and use it to detect outliers in data about coastal changes. First, they use clustering to identify the objects of interest in the spatio-temporal data. Then, the scale of data is changed to make small objects disappear. In third step, they compare (using exploratory visualization analysis) the original data with the data in the changed scaled, looking for objects that disappeared in step 2 (which are them marked as potential outliers). The last step is to check if the detected objects are outliers, comparing features of suspected objects with the ones of their temporal neighbours. We identified the presence of clusters of inaccurate data in our dataset. Then, a simple comparison of an object representation with its temporal neighbours cannot be used to identify if the considered object is inaccurate. Also, the change scale strategy cannot be applied to detect outlier-candidates among the geometries in our dataset, as it has just one geometry per timestamp.

Wu et al. [25] study spatio-temporal outlier detection over precipitation data. Their proposal is to initially find the *top-k* outliers for each time period using the Kulldorff's [17] spatial scan statistic to evaluate how discrepant a geometry is from the remainder of the data in the considered period. Then, they use the top-k outliers to divide the data into sequences, organize the sequences in trees and apply a recursive algorithm to find the outliers in the database. In our work, good quality data representing the modeled object at distinct timestamps may be considerable different from each other. On the other hand, clusters of inaccurate data can be considerable similar, and still be poor quality data we need to detect and remove.

Chandola et al. [6] and Kwon et al. [18] present surveys on anomaly and outlier detection. In a more recent work, Gupta et al [13] present a survey of outlier detection in the context of temporal data. In this work, we look for inconsistencies in the spatio-temporal evolution of the modeled phenomena as a way to detect erroneous and inaccurate data. We analyze its evolution (in terms of shape and positioning), compare geometries for distinct timestamps and verify series of metrics.

3 THE FOREST FIRE DATASET

The dataset used in our case study was built using an aerial video tracking the evolution of a controlled fire that took place at Pinhão Cel, Portugal, in 2019. It has approximately 15 minutes and was filmed using 25 fps, which leads to more than 22,500 snapshots on fire evolution. This work is part of a broader study on the estimation of gas emissions to the atmosphere.

We used video segmentation and object detection techniques to identify the burned area in each frame and generate a WKT (Well-known text) of the polygon representing the boundaries of that area. The segmented video frame images and the corresponding WKT representations are available at [21]. Then, the WKT representations of the burned area were loaded into PostgreSQL and stored using PostGIS's geometry data type. We used PostgreSQL 11, PostGIS 2.5.3 and GEOS 3.7.2.

The average number of vertices per geometry is 1,938, thus the complete dataset has more than 43,6 million vertices. While using almost 2 thousand points per geometry may indicate the objects are being represented in high detail, it may also cause performance issues while executing interpolation methods and other spatio-temporal operations. Hence, the contours are converted into vector mode using the PostGIS built-in procedure ST_SimplifyPreserveTopology [22] with a *tolerance* of 10. This procedure uses the Douglas-Peucker simplification algorithm [9] to reduce the number of vertices that represent a geometry, while maintaining a good approximation of their shape in most cases. The average number of vertices per geometry in the new dataset is only 4% of the original one, while the Jaccard Index (i.e., the ratio between the intersection and the union of two polygons: $JAC(A,B) = \frac{|A \cap B|}{|A \cup B|}$) measuring the similarity between each simplified geometry and the corresponding one in the original dataset was of 0.97 on average. This is a good trade-off between simplification and representation detail.

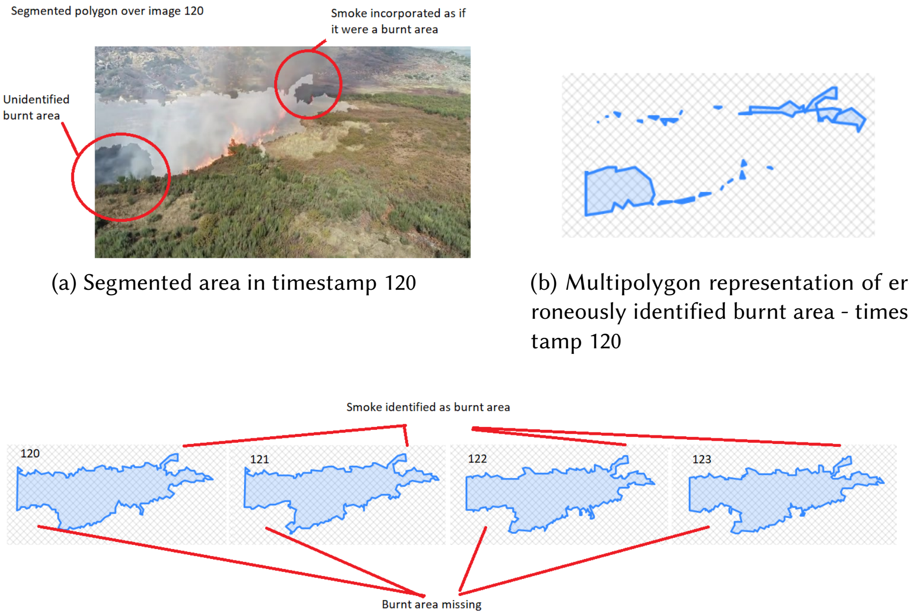
A comparison of the polygons and the original images shows that the segmentation algorithm considered smoke and burnt area as a single object in many cases, i.e., sometimes part of the smoke was included in the polygon representing the burned area, while in other cases, parts of burned area were disregarded, as if they were smoke (Figure 2). Figure 2a shows two large parts of the burned area that were not identified by the segmentation algorithm. Figure 2b displays a multipolygon representing the parts of the area identified erroneously (either missing or wrongly incorporated as burned area) for timestamp 120. This represents 23% of the polygon for the considered timestamp.

An in-depth analysis of several consecutive frames unveils the existence of clusters of outliers. For instance, consider Figure 2c, which presents the polygon for timestamps 120 to 123. The misrepresentations identified in the polygon of timestamp 120 also happen in the other ones. Hence, all of them are inaccurate. But they are consistent with each other and a simple comparison between the polygon of timestamp 121 with the previous and following is not by enough to identify that polygon as inaccurate.

4 QUALITY RULES, CONSISTENCY CHECKS AND ACCEPTANCE METRICS

As the visual inspection and adjustment of thousands of polygons is unfeasible, we propose a novel method to identify inaccurate spatial data in a time series using consistency checks and requiring just a few manually verified reference representations. The workflow has three main steps:

(1) Define consistency rules
(2) Define consistency checks and quality metrics
(3) Apply the quality assessment checks and data improvement
  (a) Define the boundaries of a time range to be evaluated
  (b) Manually refine the data for the first and last observations (i.e. polygons) on the range
  (c) Apply consistency checks to identify outliers (i.e. inaccurate data) and remove them
  (d) Check consistency acceptance requirements and go back to step 3a if necessary


Segmented polygon over image 120
Smoke incorporated as if it were a burnt area
Unidentified burnt area
Smoke identified as burnt area
120
121
122
123
Burnt area missing

(a) Segmented area in timestamp 120

(b) Multipolygon representation of erroneously identified burnt area - timestamp 120

(c) Cluster of inaccurate data - timestamps 120 to 123

Fig. 2. Examples of inaccurate data

In the following, we present the consistency rules, the assessment checks and the metrics, and discuss how to choose the data to be manually refined.

4.1 Data consistency rules

In the representation of a forest fire, we know that the area identified as burned in one observation should also be burned in all the following observations, while the area identified as not-burned should also not be considered as not-burned in any of the previous observations. Also, for any time interval ti, which begins at instant k and ends at instant l ($ti = [k, l]$), the burned area should evolve from the geometry observed at k to the one observed at l. So, considering that $P(t)$ is a polygon representing the burned area at instant t, the following equation should be valid.

$$\forall i; \forall j; k \leqslant i \leqslant j \leqslant l; P(i) \subseteq P(j) \tag{1}$$

From Equation 1, we can infer that:

(i) The function $f(t) = A(P(t))$, where A stands for the geometric area of a polygon P at instant t should increase monotonically, as defined in 2.

$$\forall i; \forall j; k \leqslant i \leqslant j \leqslant l; A(P(i)) \leqslant A(P(j)) \tag{2}$$

(ii) The function $g(t) = JAC(P(t), P(k))$, that represents the evolution of the Jaccard Index between each geometry P in ti and the first geometry in ti should start with 1 and be a monotonically decreasing function. On the other hand, the function $h(t) = JAC(P(t), P(l))$, that represents the evolution of the Jaccard Index between each geometry P in ti and the last polygon in ti, should be a monotonically increasing function that ends with 1. The lower bound for the Jaccard Index is the one computed between the first and the last observations in ti.

$$\forall i; \forall j; k \leqslant i \leqslant j \leqslant l; JAC(P(i), P(k)) \geqslant JAC(P(j), P(k)) \wedge JAC(P(i), P(l)) \leqslant JAC(P(j), P(l)) \quad (3)$$

4.2 Consistency checks

While establishing the consistency checks, we should consider that real-world data may have small variations and noise that don't necessarily make it invalid for use. For instance, in the segmentation of real world videos with fire, smoke and wind, there may exist small variations in the contours of the burned area, which may cause small inconsistencies between $P(i)$ and $P(i+1)$. Figure 3 illustrates this situation. It presents the polygons which represents burned area at the first and second observations, and the geometric difference between such polygons ($P(1) - P(2)$). According to Equation 1, such difference should be empty. But the result on the real-world data is a set of small geometries scattered along the contour of P(2). Such small inconsistency may be acceptable until the total area of the resulting multipolygon is small when compared to the burned area.

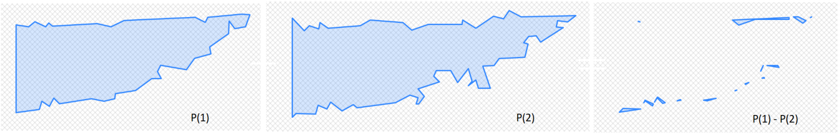


Fig. 3. Burnt area polygons for the first and second frames, and the geometric difference between them

Usually, it is possible to formulate more than one consistency check for a given dataset. Some checks are complementary to each other, Some checks are able to identify more inconsistencies than others.

Let's consider that the first ($P(k)$) and last polygons ($P(l)$) in a range (ti) are (possibly manually edited) accurate. From Equations 2 and 3 it is possible to enumerate the following criteria to identify outliers:

(C1) Any polygon in ti whose area is smaller than the one of $P(k) * \alpha$ or greater than the one of $P(l) * \beta$ is an outlier (α and β are threshold values used to deal with real-world data noise, as illustrated in Figure 3);

(C2) Any polygon in ti whose Jaccard Index with $P(k)$ or $P(l)$ is smaller than the Jaccard Index between $P(k)$ and $P(l)$ is an outlier.

Although valid, the above checks fail to identify several situations. Consider the four examples illustrated in Figure 4. Each example displays three polygons: $P(k)$ and $P(l)$, which are the accurate representations of the burned area at the first and last timestamps in a range (ti), and $P(i)$, which is an in-between representation, whose consistency must be verified.

The area checks (item (C1)) identify $P(i)$ as an outlier only in the first example. The Jaccard Index checks (item (C2)) identify $P(i)$ as an outlier in a and b. although, $P(t)$ represents an evolution of the burned area $P(i)$ that is inaccurate in all the four examples of Figure 4. So, we need another check to identify the invalid representations in c and d.

From Equation 1, we can infer that the geometric difference between the polygons $P(i)$ and $P(j)$ (i.e. $P(i) - P(j)$, $i \leqslant j$) should result in an empty polygon[1]. Therefore, considering that the first ($P(k)$) and last polygons ($P(l)$) in a range of representations (ti) are accurate, we can define another consistency check:

[1] The geometric difference between $P(j)$ and $P(i)$ is the part of the geometry $P(j)$ that does not intersects with $P(i)$

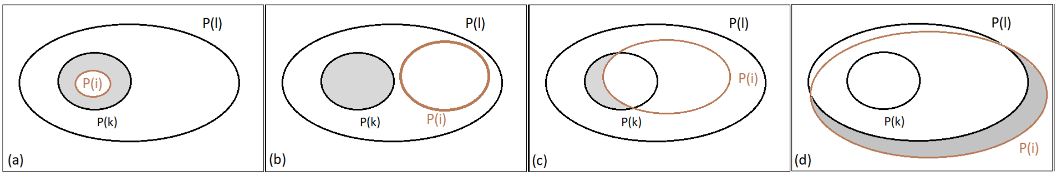


Fig. 4. Outlier configurations - Examples

(C3) Any polygon $P(i)$ in ti where the geometric differences $P(k) - P(i)$ or $P(i) - P(l)$ are not empty, is an outlier.

This geometric check (C3) would identify $P(i)$ as an outlier in all the situations represented in Figure 4 (grey filled areas represent an $P(k) - P(i)$ in images *a-c* and $P(i) - P(l)$ in image *d*). Thus, to identify if a geometry $P(i)$ is an outlier, we define the *relative inconsistency value to the initial representation* (RVI) and the *relative inconsistency value to the final representation* (RVF) as follows (Equations 4 and 5).

$$\forall i; k \leqslant i \leqslant l; RVI(i) = (A(P(k) - P(i)))/A(P(k)) \tag{4}$$

$$\forall i; k \leqslant i \leqslant l; RVF(i) = (A(P(i) - P(l)))/A(P(l)) \tag{5}$$

These geometric operations define the the representation any polygon $P(i)$ in the interval $[k, l]$ is poor when $RVI(i)$ or $RVF(i)$ are greater then a threshold θ.

4.3 Quality Acceptance and Range Definition

After applying consistency checks and outlier removal from a given time range, we execute a quality acceptance verification derived from Equation 1.

Let's consider *Dump* is a function that returns each of the polygons from a multipolygon. We define the *relative estimated error* (REE) for a polygon $P(i)$ as:

$$\forall i; \forall j; k \leqslant i < j \leqslant l \wedge P(j) \text{ is not an outlier}; REE(i) = \frac{Max(A(DUMP(P(i) - P(Min(j)))))}{A(P(i))} \tag{6}$$

Figure 3 presents a multipolygon resulting from a geometric difference operation, which is composed of several small polygons (over the contour of a geometry). This anomaly is distinct from the one in Figure 2b, which has a large polygon representing an inaccuracy. Then, to define REE, we consider the polygon with the largest area in the multipolygon resulting from the geometric difference operation. Also, in Equation 6, we use the lowest value of j (i.e. $Min(j)$) that is greater than i and in which $P(j)$ that was not marked as an outlier.

The quality acceptance test (QAT) proposed in this work considers that a representation in the interval $[k, l]$ is good if REE is smaller then a threshold δ for all the values of $P(i)$ ($k \leqslant i < l$).

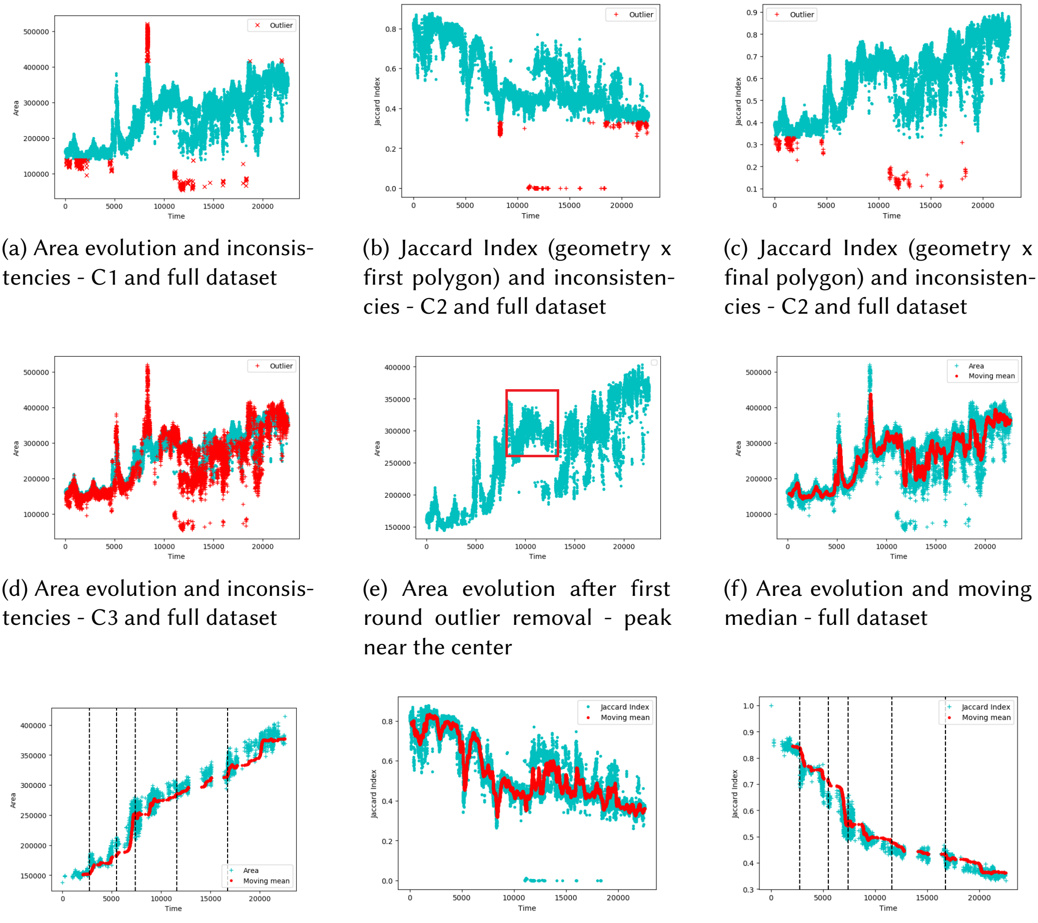
If the quality acceptance test satisfies the threshold δ, then no more cleaning is executed for that range. Otherwise, we should choose a timestamp m in $[k, l]$ and validate (and edit, if necessary) the representation of the burned area manually. After validating $P(m)$, we have two new ranges to apply the consistency checks: $[k, m]$ and $[m, l]$.

In order to choose the observation $P(m)$ to validate, we look for an instant m ($m \in [k, l]$) near a peak or valley of functions f, g or h, i.e., near where $f'(m)$, $g'(m)$ or $h'(m)$ changes its signal ($f'(t) = \frac{df}{dt}$, $g'(t) = \frac{dg}{dt}$, $h'(t) = \frac{dh}{dt}$). This point can be selected by graph analysis, as we present in the next Section.

5 EXECUTION RESULTS

The cleaning of our forest fire dataset started with the visual inspection and adjustment of the polygons for the first and last observations (i.e. first and last frames in the video). Then, we applied consistency checks C1, C2 and C3.

Figure 5d presents the area of each polygon and the inconsistencies detected using C3 (we computed RVI(i) and RVF(i) for all observations in the dataset and used $\theta = 0.05$). This check detected much more problems then C1 (Figure 5a) and C2 (Figures 5b and 5c), as we discussed in Section 4.2. Hence, in the followings refinements, we apply only check C3.



(a) Area evolution and inconsistencies - C1 and full dataset

(b) Jaccard Index (geometry x first polygon) and inconsistencies - C2 and full dataset

(c) Jaccard Index (geometry x final polygon) and inconsistencies - C2 and full dataset

(d) Area evolution and inconsistencies - C3 and full dataset

(e) Area evolution after first round outlier removal - peak near the center

(f) Area evolution and moving median - full dataset

(g) Area evolution and moving median - clean dataset

(h) Jaccard Index (geometry x first polygon) and moving median - full dataset

(i) Jaccard Index (geometry x first polygon) and moving median - clean dataset

After removing the inconsistent data detected by check C3, we applied the acceptance check defined in 4.3. Using $\delta = 0.1$, 3.2% of the acceptance tests failed. It is important to notice that the tests can only specify that there exists inconsistent data, but cannot specify which polygon is invalid. Then, we should go back to step 3a in the workflow we defined in Section 4, i.e., we should choose a new observation $P(m)$ ($m \in [k, l]$) to validate. The assessment and improvement process continued with the following range definitions and splits:

(S1) The number of range splits necessary to achieve the required quality level highly depends on the choice of the instant m. Figure 5e presents the area evolution after applying C3 and removing outliers. It also highlights a peak in the function near the middle of the interval, corresponding to good candidates to define the first range split. In the second iteration, we selected and adjusted manually a new frame (timestamp 11579), and divided the initial time series into two new ranges ($SA1 = [1, 11579], SA2 = [11579, 22532]$). Then, we applied the consistency check C3 (we now use $\theta = 0.1$, since any small error in fire propagation representation will remain in initial/final boundaries) over these two sets and removed inconsistent data. The results of QAT for $SA1$ and $SA2$ were 1.1% and 0.05%, respectively, and so the test fails in both cases.

(S2) We manually adjusted two new images and split each of the intervals into two new ones, creating the sets: $SB1 = [1, 5499]$; $SB2 = [5499, 11579]$, $SB3 = [11579, 16751]$ and $SB4 = [16751, 22532]$. After applying C3 and removing outliers, the fails rate for QAT was of 0.7% and 0.9% for $SB1$ and $SB2$. In $SB3$ and $SB4$, all the values of QAT were acceptable and no more refinement is required for them;

(S3) To continue cleaning $SB1$ and $SB2$, we manually refined two new observations and created the ranges $SC1 = [1, 2763]$, $SC2 = [2763, 5499]$, $SC3 = [5499, 7393]$ and $SC4 = [7393, 11579]$. After applying C3 check, only $SC1$ failed QAT tests. To avoid further splits, we applied C3 check with $\theta = 0.05$ to the points in the range $SC1$ (removing some more inconsistent data) and it passed the QAT test.

Then, after manually adjusting 7 polygons, we got a consistent set composed by 3613 polygons (16% of the original dataset). Figures 5g and 5i present the area evolution and Jaccard Index evolution for the cleaned dataset. The dashed lines represent the timestamps of the polygons that were visually inspected. Although there are some local small inconsistencies (whose amplitude relies on the used values of θ and δ) in the area and Jaccard Index evolution, the global tendency is totally consistent (which can be verified by the represented moving means). Also, when comparing such metrics evolution on the clean dataset with the ones on the original dataset (Figures 5f and 5h), it is possible to notice the effectiveness of used methods.

Although the final dataset contains just about 16% of original data, such number of polygons corresponds, in average, to 4 representations of real data per second. This is totally acceptable in the context of the studied phenomena. Also, as the overall data quality was improved, any required in-between representation can be obtained using existing region interpolation functions.

6 CONCLUSIONS

Efficient use of real data about moving regions is still an open issue, as real-world data is usually subject to noise and anomalies that are often difficult to identify and remove. In this work, we present the Forest Fire dataset, composed of thousands of polygon representations of the evolution a real world phenonema, and describe our strategy to improve the overall quality of the dataset.

As the actual evaluation of the accuracy of each polygon depends on visual inspection, we created a set of consistency rules, checks and quality acceptance metrics to guarantee data consistency. We validate the accuracy of just a few polygons and apply geometric operations on the remaining ones until desired consistency levels are achieved.

The proposed strategies can be applied to other real-world datasets, specially the ones representing the evolution of real world phenomena, like the retreat of glaciers and the route of lava flows. As future work, we plan to apply the strategy to other datasets of our studies on gas emission and burned area evolution. We also plan to apply the used workflow to real-world datasets related to cells' life cycle.

ACKNOWLEDGMENTS

This work is partially funded by National Funds through the FCT (Foundation for Science and Technology) in the context of the projects UID/CEC/00127/2013 and POCI-01-0145-FEDER-032636.

REFERENCES

[1] N. R. Adam, V. P. Janeja, and V. Atluri. 2004. Neighborhood Based Detection of Anomalies in High Dimensional Spatio-Temporal Sensor Datasets. In *Procs of the 2004 ACM Symposium on Applied Computing (SAC 04).* 576–583.

[2] C. C. Aggarwal, Y. Zhao, and P. S. Yu. 2011. Outlier detection in graph streams. In *2011 IEEE 27th International Conference on Data Engineering.* 399–409.

[3] Ira Assent, Philipp Kranen, Corinna Baldauf, and Thomas Seidl. 2012. AnyOut: Anytime Outlier Detection on Streaming Data. In *Database Systems for Advanced Applications.* Springer Berlin Heidelberg, 228–242.

[4] Ana Bianco, Marta Garcia Ben, E. MartÃŋnez, and Victor Yohai. 2001. Outlier Detection in Regression Models with ARIMA Errors Using Robust Estimates. *Journal of Forecasting* 20, 8 (December 2001), 565–579.

[5] D. Birant and A. Kut. 2006. Spatio-temporal outlier detection in large databases. In *28th International Conference on Information Technology Interfaces.* 179–184.

[6] V. Chandola, A. Banerjee, and V. Kumar. 2009. Anomaly Detection: A Survey. *ACM Comput. Surv.* 41, 3 (2009), 1–58.

[7] Tao Cheng and Zhilin Li. 2006. A Multiscale Approach for Spatio-Temporal Outlier Detection. *Transactions in GIS* 10, 2 (2006), 253–263.

[8] Ciro D´ Urso. 2016. EXPERIENCE: Glitches in Databases, How to Ensure Data Quality by Outlier Detection Techniques. *J. Data and Information Quality* 7, 3 (Sept. 2016).

[9] D Douglas and T Peucker. 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: the intern. journal for geographic infor. and geovisualization* 10, 2 (1973), 112–122.

[10] J Duarte, P Dias, and J Moreira. 2018. A Framework for the Management of Deformable Moving Objects. In *Geospatial Technologies for All.* Springer, 327–346.

[11] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *KDD.*

[12] L: Forlizzi, R. Güting, E. Nardelli, and M. Schneider. 2000. A Data Model and Data Structures for Moving Objects Databases. In *Procs. of the 2000 ACM SIGMOD International Conference on Management of Data (SIGMOD '00).* 319–330.

[13] Manish Gupta, Jing Gao, Charu C. Aggarwal, and Jiawei Han. 2014. Outlier Detection for Temporal Data: A Survey. *IEEE Trans. Knowl. Data Eng.* 26, 9 (2014), 2250–2267.

[14] D J. Hill and B S. Minsker. 2010. Anomaly Detection in Streaming Environmental Sensor Data: A Data-Driven Modeling Approach. *Environ. Model. Softw.* 25, 9 (2010), 1014–1022.

[15] J.N. Hrid and Greg Mcdermid. 2008. Noise reduction of NDVI time series: an empirical comparison of selected techniques. *Remote Sens. Environ.* 113 (01 2008), 248–258.

[16] S. Jiang and Q. An. 2008. Clustering-Based Outlier Detection Method. In *2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery*, Vol. 2. 429–433.

[17] Martin Kulldorff. 1997. A Spatial Scan Statistic. *Communications in Statistics - Theory and Methods* 26 (06 1997), 1481–1496. https://doi.org/10.1080/03610929708831995

[18] Donghwoon Kwon, Hyunjoo Kim, Jinoh Kim, Sang C. Suh, Ikkyun Kim, and Kuinam J. Kim. 2019. A survey of deep learning-based network anomaly detection. *Cluster Computing* 22, 1 (01 Jan 2019), 949–961.

[19] Jinbo Li, Witold Pedrycz, and Iqbal Jamal. 2017. Multivariate Time series Anomaly Detection: A Framework of Hidden Markov Models. *Applied Soft Computing* 60 (06 2017).

[20] Mark McKenney and James Webb. 2010. Extracting Moving Regions from Spatial Data. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '10).* ACM, 438–441.

[21] MoST-Team. 2020. MoST Forest Fire Dataset. Retrieved February, 2020 from http://most.web.ua.pt/Video15minImagesWKT.zip

[22] PostGis-Team. 2020. ST_SimplifyPreserveTopology - PostGIS dev Manual. Retrieved February, 2020 from https://postgis.net/docs/ST_SimplifyPreserveTopology.html

[23] Erlend Tøssebro and Ralf Hartmut Güting. 2001. Creating Representations for Continuously Moving Regions from Observations. In *Advances in Spatial and Temporal Databases.* Springer Berlin Heidelberg, 321–344.

[24] R. S. Tsay, D. Peña, and A. E. Pankratz. 2000. Outliers in multivariate time series. *Biometrika* 87, 4 (2000), 789–804.

[25] Elizabeth Wu, Wei Liu, and Sanjay Chawla. 2008. Spatio-Temporal Outlier Detection in Precipitation Data. In *Proceedings of the Second International Conference on Knowledge Discovery from Sensor Data (Sensor-KDD).* Springer-Verlag, 115–133.