Creating Moving Region Representations: a Case Study on the Spread of a Forest Fire

Henrique Macías da Silva (macias66@ua.pt)
Tiago F. R. Ribeiro (tiago.f.ribeiro@ipleiria.pt)
Rogério Luís de C. Costa (rogerio.l.costa@ipleiria.pt)
José Moreira (jose.moreira@ua.pt)

(2023)

Abstract *This work focuses on generating spatiotemporal data from real-world observations to represent the evolution of phenomena of interest as moving regions. The case study is the creation of a dataset to represent the spread of a controlled forest fire from aerial images captured using a drone. We present an overview of the data acquisition and preparation steps, and describe the optimization strategy implemented to establish a vertex correspondence between the regions that delimit the burned area at discrete time instants. These data can be used to create a continuous representation of the evolution of the burned area over time.*

1 Introduction

There is a lot of work on data models and spatiotemporal query languages. Creating spatiotemporal data such as deformable moving regions, i.e., regions that may change position, orientation, size and shape continuously over time, is challenging and time-consuming. In addition, it is often difficult to transfer the solutions prepared for a use case to another. Therefore, the vast majority of the experimental work on representing, querying and analyzing spatiotemporal data uses synthetic data, created manually to test specific conditions or created using generators. However, using real-world data is also important, not only because they highlight issues that are not considered when creating or generating synthetic data, but also to learn how to develop methods and tools to create good quality data to be used in real-world applications.

In this work, we present the methods used to create a dataset representing the spread of a controlled forest fire. First, we present an overview of previous work to obtain the areas of interest from raw data, in this case, a video captured using a drone, remove outliers and select good snapshots to represent the evolution of the burned area over time. The output consists of a data series where each entry is composed of a timestamp and a region (polygon) representing the

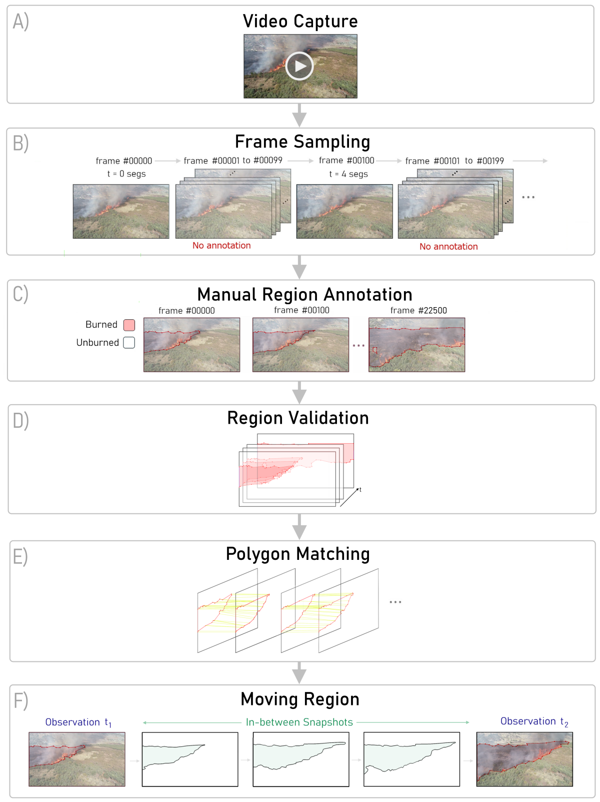

Figure 1: Dataset Generation Workflow

burned area at that time. Second, we present the methods used to create vertex correspondences between two consecutive regions in the data series as proposed in [Liu et al.(2004)]. Unlike the examples in [Liu et al.(2004)], the regions used in this work are very irregular, and therefore they usually have many vertices, and the deformation is highly variable, so, it is not possible to find a set of parameters usable for all entries in the data series. Thus it was necessary to devise an approach to find these parameters dynamically. Third, we present the details of the dataset created in this work and some samples to illustrate the results and the issues found in this work. Finally, we present a discussion of the results and lessons learned during this work and highlight possible applications and guidelines for future work.

2 Dataset Generation

The dataset presented in this work was derived from videos captured during prescribed fires conducted in Portugal. Building upon these videos, a series of successive steps were undertaken, as illustrated in Figure 1, which are elaborated upon in this section. Next, the methods used to produce the datasets are presented.

2.1 Data Capturing and Segmentation

Capturing adequate quality drone video for the study of the burned area evolution of a hotspot with a drone is a challenging process in which planning is imperative to mitigate any potential hazards. The selection of the location needs to take into consideration the local orography, accessibility, or the type and amount of fuel on the site. Wind speed, temperature and relative humidity at the time of the capture are also other key factors to consider. The data that serve as the basis for the dataset, were collected during a campaign of prescribed fires whose purpose was to prevent excessive accumulation of combustible organic matter and reduce the risk of wildfires in northern Portugal. Specifically, the video was recorded by a drone equipped with an RGB camera during a prescribed fire at Torre do Pinhão (41° 23' 37.56", -7° 37' 0.32") on the 1st of March 2019, starting around 12:20 pm. The resulting video spans approximately 15 minutes, with a frame rate of 25 fps and a resolution of 720×1280, amounting to 22500 images. Throughout the data collection, the drone remained nearly stationary. However, despite efforts, there is considerable occlusion of the area of interest in several parts of the video, mainly due to smoke.

Prior to annotation, video frames are extracted, maintaining the original frame resolution. Subsequently, periodic sampling is conducted, discarding intermediate frames. For this application, we sample every 100th frame, which corresponds to a period of 4 seconds. The use of automatic segmentation methods led to regions being noisy and susceptible to occlusions by fire and smoke. To tackle these issues, we opted to manually annotate the frames. To minimize subjectivity, we define `burned area` as the entire area (surface of the terrain) affected by the fire. In some instances, smoke and flames introduce occlusions that pose challenges in accurately segmenting certain video frames. To remedy this difficulty, we define that if it is not possible to be sure that a certain area was consumed by fire, it is not considered a burned area. Additionally, as a measure of last resort, we use previous frames of the video to establish the boundaries of the burned area, always conservatively.A detailed description of manual segmentation is in [Ribeiro et al.(2023a)]. The manual annotation resulted in a total of 249 annotated frames. Out of these, 226 frames span from the initial frame to the end frame (#22500), while the remaining 23 regions are offset by 50 frames, starting from frame #20250 and ending at frame #22450.

The original MP4 video, segmentation mask-frame pairs in PNG and JSON format, WKT segmentation mask polygons, metadata from the drone (positioning, height, GPS coordinates, orientation, and camera sensor details) during data collection, as well as high-resolution orthophotographs of the area before and after the prescribed fire area, are can be found at [Ribeiro et al.(2023b)]. After manual segmentation, data validation methods were applied. considering the validation methods proposed in [Costa et al.(2021)].

2.2 Region matching

Let $\bar{S} = S_1, S_2, \ldots, S_m$ be a sequence of snapshots on the evolution of the shape of an object or event over time. Each snapshot is a tuple $S_j = (t_j, R_j)$ such that t_j is a timestamp and $R_j = p_1, p_2, \ldots, p_{n_j}$ is a region with n_j vertices denoted as p_i, with $i = 1, 2, \ldots, n_j$. The timestamps are ordered $(t_j < t_{j+1}, j = 1, 2, \ldots m-1)$ and each region must follow the rules commonly established for the representation of spatial data in databases and Geographical Information Systems (GIS), namely, the coordinates of the first and last vertices are equal and the edges cannot intersect. The number of vertices n_j of each region can vary and the regions cannot have holes. In the following, for any pair of consecutive regions, R_j is denoted as source (S) and R_{j+1} is denoted as target (T).

Next, we define a mapping function $M = map : R_j \mapsto R_{j+1}$ to establish a one-to-one correspondence between the vertices of S and T. Since the number of vertices in S and T most likely differs, the mapping function must add vertices to the exterior of S and T to ensure a one-to-one correspondence as presented in Section 2.2. To preserve the topology of the polygon during transformation, the relative position of the vertices in S and T must be kept as depicted in Figure 2. In this example, two clones of vertex number 2 were added in T, but other approaches are also possible.

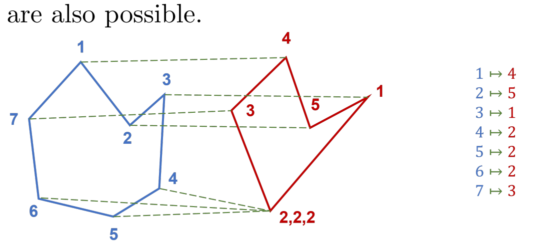

Figure 2: Vertex correspondences between a source (blue) and target (red)

To obtain a continuous representation over time, an interpolation function ($MR = interp : M$) is used, such that M denotes the vertex correspondences between R_j and R_{j+1}, and MR is a moving region. We also define a function ($R = projection : MR, t$) that returns an estimation of the geometry (R) of a moving region (MR) at a time instant (t). Ideally, the transformation should approximate the evolution of the object or event of interest as closely as possible, but choosing the best interpolation method for a given use case is beyond the scope of this work. To obtain a continuous representation for the whole time interval covered by $\bar{S}$, i.e., $[t_1, t_m]$, $m-1$ interpolations are needed, hence $m-1$ mappings are also needed.

The approach used in this work to compute the vertex correspondences between source and target regions was proposed in [Liu et al.(2004)]. The algorithm starts by selecting a subset of the vertices of S and T, the so-called feature points, computes the correspondences between feature points and then computes the correspondences between the remaining vertices.

2.2.1 Geometric properties of feature points

At the beginning of the algorithm, all vertices of a region are candidates for feature points and their screening is done based on the following rules:

$$\begin{gathered} d_{min} \leq \|p_i - p_i^+\| \leq d_{max} \\ d_{min} \leq \|p_i - p_i^-\| \leq d_{max} \\ \alpha \leq \alpha_{max} \end{gathered} \tag{1}$$

where d_{min}, d_{max} and α_{max} stand for the minimum distance, maximum distance and maximum angle, p_i is the candidate vertex, p_i^- is a previous (counter-clockwise) candidate vertex and p_i^+ is a subsequent (clockwise) candidate vertex.

Each feature point candidate has a Region of Support (ROS) composed of the feature point itself and its neighbors (p_i^-, p_i, p_i^+). In order to determine the curvature of the ROS, the eigenvectors and eigenvalues are calculated through its covariance matrix, taking into account all vertices between p_i^- and p_i^+, and are then used to compare it with the ROS of other polygons. In this work, we use the measure and cost functions proposed in [Liu et al.(2004)], sections 3.1 and 3.2, and so their definition is omitted. Figure 3 shows an example of the result obtained after calculating the feature points.

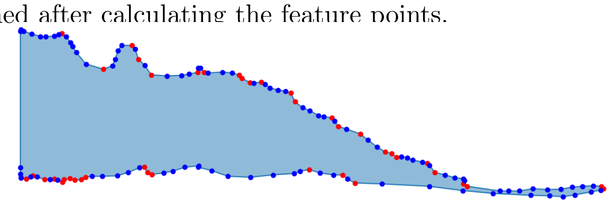

Figure 3: Candidates (blue) and chosen feature points (red)

2.2.2 Correspondence Problem

The first step is to compute the correspondences between the feature points of S and T. During the execution of the algorithm, the correspondences are represented as a $map : fS_{a+k_1} \mapsto fT_{b+k_2}$, where fS_{a+k_1} and fT_{b+k_2} are feature points and k_1 and k_2 denote the number of skips in each feature points list. The latter are needed because the number of feature points in S and T differs and so not all feature points have a correspondence. The indices are treated as a circular list.

After the correspondences between feature points are computed, the correspondences between the other vertices are added to the solution as depicted in Figure 4. The mappings are given by the formula $q_b \mapsto p_{\lfloor b \cdot \frac{ns}{nt} \rfloor}$, where ns and nt denote the number of vertices to be mapped in S and T, b denotes the vertex positions, and so the mappings are $q_0 \mapsto p_0$, $q_1 \mapsto p_0$ and $q_2 \mapsto p_1$. The same holds when the number of vertices in S is greater than in T.

At this point, we have all correspondences between the vertices of any pair of regions S and T in $\bar{S}$ and we are ready to use an interpolation method based on vertex correspondences to estimate the geometry of the region at any time instant in $[t_1, t_m]$.

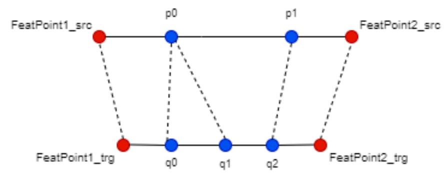

Figure 4: Intermediate correspondences

2.2.3 Dynamic parameter setting

The question that arises is which values of d_{min}, d_{max} and α_{max} in (1) should be used? From experience, we have found that no predefined values can be applied to all cases because our data is very irregular. Therefore, we need an approach to set up automatically these parameters for every pair of consecutive regions in a data series.

This is an optimization problem and, to start with, we defined the following ranges based on the results of initial experiments: $\alpha_{max} \in [120, 180]$ with step 10, $d_{min} \in [0, 20]$ with step 1 and d_{max} is equal to the length of the longest edge of the region. We also need a cost function to validate the whole set of matches between S and T. In this work, we consider the following measures.

- Average distance (AD). Measures the average of the distances between the vertices in S and the corresponding ones in T. For instance, in Figure 5, this measure is equal to the average length of the line segments in blue.

- Improved average distance (IAD). This is a version of AD where the cost of a correspondence depends on the shortest distance of a vertex to the other region:

$$IAD = \frac{1}{n'} \frac{\sum_{j=1}^{n'} \|p_j, q_j\|}{max(\|p_j, T\|,\ \|S, q_j\|)} \tag{2}$$

  such that p_j and q_j are corresponding vertices in S and T, respectively, n is the number of vertex correspondences and $\|x, y\|$ is a distance between two vertices or between a vertex and a region.

- Jaccard Index (JI). Given a source R_j, a target R_{j+1}, a mapping $M = map : R_j \mapsto R_{j+1}$, a moving region $MR = interp : M$, a time instant $t = (t_j + t_{j+1})/2$ and the projection of MR at t denoted E, the Jaccard Index is:

$$JI = \frac{1}{2} \frac{(R_j \cap E) + (R_{j+1} \cap E)}{R_j \cup E} \tag{3}$$

  This is a measure of the similarity between the region E that is halfway between the source and target, and the source and target themselves.

- Combined Jaccard Index (CJI). This is a combination of IAD and JI, i.e., $CJI = IAD \times JI$.

The transformation of the moving regions is not homogeneous and there are parts that change more than others. For instance, the burned area evolves from left to right and therefore the distance between the corresponding vertices tends to be greater for the rightmost vertices. This is most important when the time interval between snapshots is large because distances tend to increase and often mask bad matches in other parts of the region when averaging distances. This is why defined IAD in addition to AD.

For optimizing the process, some rules are defined at the level of jump conditions and exceptions. If it takes more than a predefined time to get the feature points it means that too many feature points are being obtained and so a jump condition is executed and the next loop iteration begins. As for exceptions, they occur when not enough feature points are found for the given α_{max}, d_{min} and d_{max} and a jump to the next iteration is also performed.

3 Dataset Description and Evaluation

To present the results of this work we use the manually annotated dataset presented in Section 2.1. We selected the first 195 out of the 225 initial samples (Pol195) because the remainder were regions with holes, a problem not addressed in this work. We also select a subset of Pol195 with 11 snapshots (Pol11) corresponding to entries 1, 20, 29, 49, 57, 75, 83, 88, 107, 129, 174. Thus, we have two representations of the same event with different temporal granularity.

The procedure to compute the vertex correspondences was applied twice for each data series, the first using the JI and the second using the CJI as measures. The measures AD and IAD were discarded because the results of preliminary experiments were poor. Figure 5 displays the vertex correspondences between regions 4 and 5 of Pol11. The Jaccard Index was used to find the best combination of the input parameters.

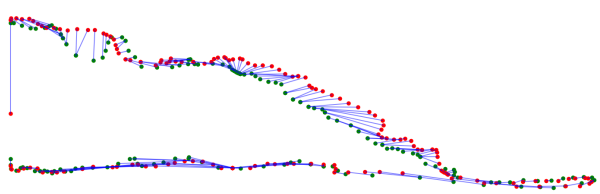

Figure 5: Vertex correspondences between a source (green) and a target (red).

For simplicity, the algorithm chosen to create the interpolations assumes that the movement of the vertices is linear and uniform. This algorithm was used to create the interpolations for the complete datasets. A score, ranging from 1 to 5 with 1 being very bad and 5 very good, was assigned to each transformation between source and target (Table 1).

The tests performed on the dataset Pol11 generated good matches for all regions with a single run, using either JI and CJI. The second set of tests was

Table 1: Datasets results

Measure	Dataset	Score	Num. Corr.	5	4	3	2	1
JI	Pol11	4.30	10	5	3	2	0	0
CJI	Pol11	4.40	10	6	2	2	0	0
JI	Pol195	4.14	170	89	33	38	3	7
CJI	Pol195	3.93	170	70	44	37	12	7

performed on the dataset with 195 regions, and it was necessary to apply the algorithm twice because a solution could not be found for several cases, e.g., when the number of feature points was not enough to generate the vertex correspondences. For instance, the vertex correspondences between regions $85 \mapsto 86$ and $86 \mapsto 87$ could not be found in the first run. So, region 86 was removed from the data series and in the second run, a solution was sought for the correspondences between $85 \mapsto 87$. In addition, the first five transformations were considered poor and regions 2–5 were discarded. So, the first mapping in the resulting dataset is $1 \mapsto 6$. The resulting dataset consists of 171 regions, meaning that 24 polygons were discarded due to no matches or very low JI values. Note that even discarding several entries in the dataset we still can have a representation without gaps between the first and last snapshots. We are only eliminating low-quality entries and, given the continuity of the model, the quality of the global representation tends to increase.

The biggest setback found has to do with the crossing of correspondences. This happens when the algorithm finds a ROS in the target that is very similar characteristics to a ROS in the source but is in very a different position. However, these cases do not occur often and, as illustrated in Table 1, the final score for each dataset is around 4 indicating good matches in most cases. The punctuation 1 and 2 represent the cases in which the vast majority of matches are cross-matched, 1 being for almost all and 2 for a good portion. 3 represents those in which some matches are not good, but in which the final result is acceptable, 4 is used when most matches are good, and 5 is used when almost all matches are good.

The data, code and materials produced in this work are available online (click to jump to Github). We also include the original video, 4 videos showing the interpolations mentioned in this section, and high-resolution aerial photographs before and after the burn.

4 Summary

This work presents the acquisition and preparation of a dataset representing the evolution of a controlled forest fire. This task covers several steps and the main focus is to find the vertex correspondences between regions representing the geometry of the burned area at different times and then apply an interpolation method to represent their evolution during a time interval. These data can be used to visualize the evolution of the moving regions, in this case representing the burned area, with no gaps in time, or to compute numerical properties regarding the evolution of the region.

Creating spatiotemporal datasets can also play an important role in benchmarking. While there is a large amount of data about moving points, the same is not true for moving regions and moving lines. From our experience, transforming raw data into moving regions is very use-case dependent and requires user supervision, at least in part. Creating many datasets, with lots of data, could enable training machine learning methods for transforming raw data into spatiotemporal data and to develop generic methods that could be used in many use cases instead of using special-purpose methods, as is currently the case.

References

[Costa et al.(2021)] Rogério Luís C. Costa, Enrico Miranda, Paulo Dias, and José Moreira. 2021. Experience: Quality Assessment and Improvement on a Forest Fire Dataset. *J. Data and Information Quality* 13, 1, Article 5 (jan 2021), 13 pages. https://doi.org/10.1145/3428155

[Liu et al.(2004)] Ligang Liu, Guopu Wang, Bo Zhang, Baining Guo, and Heung-Yeung Shum. 2004. Perceptually based approach for planar shape morphing. In *12th Pacific Conference on Computer Graphics and Applications, 2004. PG 2004. Proceedings.* 111–120. https://doi.org/10.1109/PCCGA.2004.1348341

[Ribeiro et al.(2023a)] Tiago F. R. Ribeiro, Fernando Silva, José Moreira, and Rogério Luís C. Costa. 2023a. Burned area semantic segmentation: A novel dataset and evaluation using convolutional networks. (2023). https://doi.org/10.1016/j.isprsjprs.2023.07.002

[Ribeiro et al.(2023b)] Tiago F. R. Ribeiro, Fernando Silva, José Moreira, and Rogério Luís C. Costa. 2023b. BurnedAreaUAV Dataset (v1.1). https://doi.org/10.5281/zenodo.7944963