

Extraction and Analysis of Sport Activity Data Inside Certain Area

Luka Lukač

University of Maribor,
Faculty of Electrical
Engineering and Computer Science,
Koroška cesta 46, 2000 Maribor, Slovenia
luka.lukac@student.um.si

ABSTRACT

Nowadays, sport data analysis is one of the crucial factors, used to enhance the athletes' performance, which can depend upon many different circumstances. One of those is the area of an exercise, which can dramatically impact on an athlete's performance. Since not enough devotion has been given to this topic, this study focuses on extracting and analysing parts of exercises, which take place inside of a specific area, using principles from another part of Computer Science, Computational Geometry.

Keywords sports, area, extraction, analysis

1 Introduction

As long as professional sport competitions have been organised, professional, as well as amateur athletes have been constantly trying to enhance their performance. In order to know in which aspects their exercises can be improved, tracking and analysing data are crucial. One of important factors, which can influence the performance, is the area where the training is performed. This study focuses on the extraction of the data inside certain areas, which allows thorough analysis of the effectiveness of an athlete in those regions.

2 Materials and Methods

2.1 Materials

In these days, a lot of athletes track their performance during the training using modern sport trackers, which allow them to capture a lot of indicators of their exercise. Since there are many indicators that are not visible directly, but can be extracted as a product of an extensive data analysis, a sport activity can be analysed more thoroughly and deeply. A great tool in analysing the data is Machine Learning, efficient computational methods of which enabled rising the researches in automatic planning of sport training sessions. [1]

Nowadays, many athletes monitor their performance using modern sport equipment, such as sport watches or sport wrists. There are websites that are focused on collecting that kind of data, however, it is very likely that the data does not have a public access [3] and that the

databases require a lot of preprocessing skills. An example of a toolbox for extracting features from a sport activity is 'sport-activities-features', which is available in GitHub repository at: <https://github.com/firefly-cpp/sport-activities-features>. It is focused on one of the most intensive and demanding tasks in Machine Learning: data preprocessing. [2]

The overall load indicators that are parts of sport activity datasets (total distance, total duration, average heart rate, etc.) have some disadvantages, such as that details are not expressed sufficiently, only a general outlook of the training is captured and different phases of the training are not recognized directly or are not recognized at all. Thus, more hidden indicators, like intervals, topographic maps and weather data are extracted in the 'sport-activities-features' toolbox. [2]

One of the crucial features that allows retrieving a lot of interesting and useful data is area detection, which is the main part of this study. It is well known that doing exercises in hilly areas can be a lot more difficult and demanding. Despite apparently worse performance in those regions (if only overall indicators are observed), an athlete can put more power in that activity compared to the one which is set in a flat area. As different areas may contain distinct characteristics, it is important to be able to analyse the performance of an athlete in a certain area.

Datasets that are used to extract this kind of data must include positions of an athlete at a certain time. If a dataset is corrupt or some data is missing, the analysis can output wrong results, therefore it is very important to have datasets that have no such issues.

2.2 Methods

In order to unambiguously detect points that are located inside the chosen area, inclusion test is a necessity. There are three major algorithms one can use to determine, whether a point is inside the area:

- Algorithm of the same signs,
- Algorithm of the sum of angles,
- Algorithm with rays.

To start with, the first algorithm that was implemented and seemed to be working was the algorithm of the same signs. Its principles are quite simple: if the sign of the cross product between the hull line and the line from



the hull to the point is the same for all the hull lines, the point lies within the polygon that represents the area with great certainty. However, there are two significant disadvantages regarding this algorithm, namely, it does not allow holes inside the area, and secondly, it does not allow concave polygons (representing area). [4]

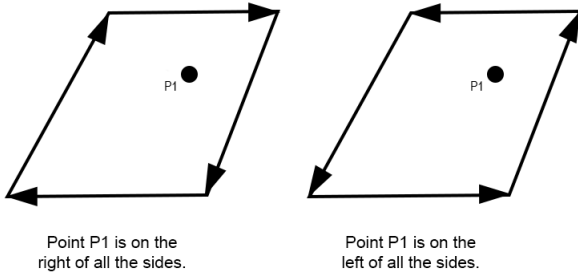


Figure 1: Algorithm of the same signs.

The next algorithm that was considered was the algorithm of the sum of angles. To each point that is located on the hull, a ray from the point, which has been chosen for the inclusion test, is made. The next step is to calculate the sum of the angles between the rays, constructed in the previous step. If the sum is 360° , the point is in the area and if the sum is 0° , the point lies outside the area. Although this algorithm allows concave angles on the border of the area, the holes are still not permitted, this is why it has not been implemented to the package as an inclusion test. [4]

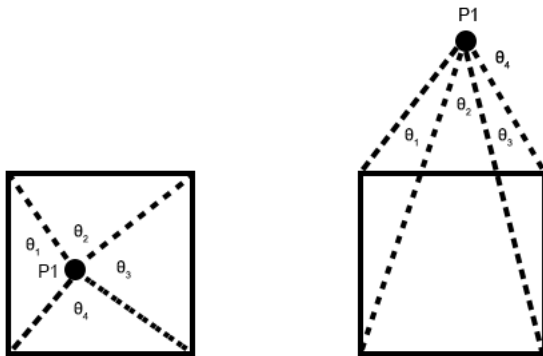


Figure 2: Algorithm of the sum of angles.

Finally, the third algorithm that uses rays to determine whether a point lies inside an area has been implemented. A ray from the point, which we want to check if inside or outside the area, to any direction is being constructed. If the ray intersects with the area border even times, the point is not a part of the area. On the other hand, if there is an odd number of intersections between the ray and the hull, the point lies inside the area. This algorithm is much less error-prone than the aforementioned algorithms, as

it produces the right results even if there are holes or concave angles present in the area, this is why this algorithm has been chosen for determining which parts of an exercise were inside the given area. [5]

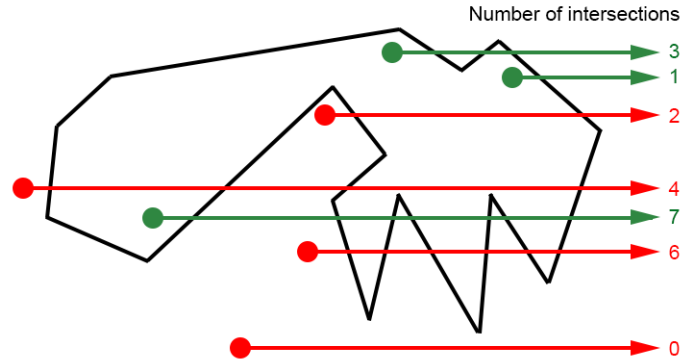


Figure 3: Ray algorithm.

```

1 function GET-POINTS-INSIDE-AREA(positions,
  area)
2   points_in_area ← {};
3   for position ← 1 to positions.count do
4     n ← number_of_intersections(position,
  area);
5     if n % 2 = 1 then
6       points_in_area ← points_in_area +
  position;
7     end
8   end
9   return points_in_area;

```

Algorithm 1: Detection of points of an exercise inside the given area

3 Results

After the implementation of the algorithm that detects parts of an exercise inside area, visualisation has been made in order to be able to test the algorithm properly. To start with, the map sectors, where the exercise is taking place, are downloaded from OSM (OpenStreetMap) and merged into an image, represented as a map. Then, the area that has been used for the extraction of data is plotted on the map, and lastly, the identified and unidentified points of an exercise are being drawn on the map according to the coordinates. An example of the visualization can be found at 5, where the wider area of the Slovenian Littoral is being displayed on the map and the exercise is plotted.

However, not only are parts inside the area extracted, but the data inside the area is also analysed. As already pointed out, the performance of an athlete inside different areas can vary significantly. This is why it is important to be able to retrieve data, specific to a certain region. One important factor about areas is also, what the performance of an athlete was like not only in one training, but

in more exercises that took place at different times. This is why data, such as whole distance, average speed, maximum speed, average heart rate, etc. is being extracted from the activities or their parts, which took place inside the same area.

Despite the algorithm working flawlessly, there can be issues with obtaining datasets which would suit the analysis. As this kind of data is usually not publicly available, a user is often limited to analysing their own activities, which is one of the reasons why there is only a limited amount of exercises visualized and displayed in this paper.



Figure 4: Visualization of the area detection.

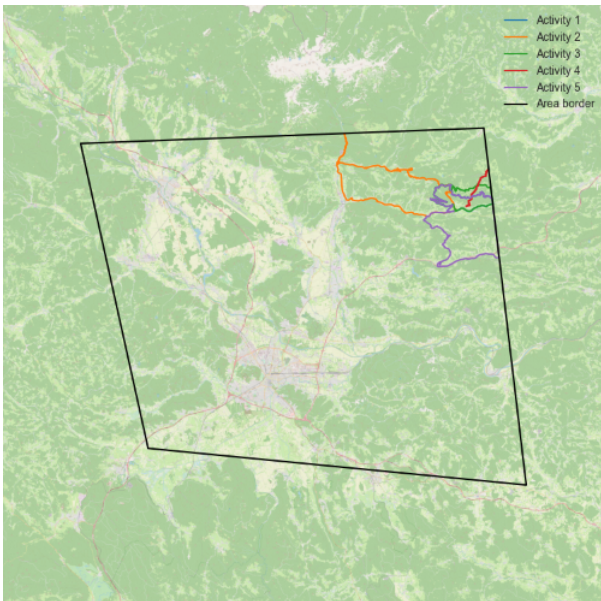


Figure 5: Visualization of more exercises inside area.

4 Discussion

To conclude, area identification is one of the crucial factors in the sport data analysis. Since different areas have different difficulties, it is crucial to be able to analyse regions separately. This study portrays how this task can be done using algorithms from Computational Geometry. In the beginning, parts of exercises inside desired areas should be extracted and then the extracted data can be used for analysis of the performance of an athlete. The next step, which has not been a part of this study, could be the Machine Learning phase, during which the extracted data could be analysed even further.

Acknowledgment

The author of this paper would like to kindly thank Iztok Fister Jr., who provided a lot of motivation and assistance during the writing of this study.

References

- [1] FISTER, I., FISTER JR, I., AND FISTER, D. *Computational intelligence in sports*. Springer, 2019.
- [2] FISTER JR., I., LUKAČ, L., RAJŠP, A., FISTER, I., PEČNIK, L., AND FISTER, D. A minimalistic toolbox for extracting features from sport activity files. In *2021 IEEE 25th International Conference on Intelligent Engineering Systems (INES) (2021)*, IEEE, pp. 121–126.
- [3] RAJŠP, A., AND FISTER JR., I. A systematic literature review of intelligent data analysis methods for smart sport training. *Applied Sciences* 10, 9 (2020), 3013.
- [4] TOPIWALA, A. Is the point inside the polygon? *Towards Data Science* (2020).
- [5] VACEK, L., ATTER, E., RIZO, P., AND NAM, B. suas for deployment and recovery of an environmental sensor probe.

Table 1: List of extracted features inside area.

ID	Feature	Description
1	Distance	Total distance inside area.
2	Time	Total time inside area.
3	Maximum speed	Maximum speed inside area.
4	Average speed	Average speed inside area.
5	Minimum heart rate	Minimum heart rate inside area.
6	Maximum heart rate	Maximum heart rate inside area.
7	Average heart rate	Average heart rate inside area.

