# A Heuristic Approach to Merging Spatial Datasets of the Bulgarian Cultural and Historic Heritage

Alexander Petkov[1, 2]

[1]*Informatics Department, Faculty of Applied Mathematics and Informatics, Technical University of Sofia, Bulgaria*
[2]*Research and Development Sector, Technical University of Sofia, Bulgaria*

Corresponding author: alex@acstre.com

**Abstract.** Digitalization of different historic maps of the same area inevitably leads to different coordinates for the same objects due to map accuracy, digitalization techniques and other factors. This paper presents a heuristic approach to merging nearby objects from different datasets into one when objects are sparsely distributed relative to each other. The approach relies on general-purpose software that is readily available on many personal computers or can be installed free of charge.

## INTRODUCTION

The cultural and historical heritage is an important part of every nation's history. On August 17, 2018, the Council of Ministers of The Republic of Bulgaria approved the "Cultural Heritage, National Memory and Social Development National Research Program" (Council of Ministers of The Republic of Bulgaria, 2018). The program has the following specific goals:

1. Development of digital instruments for research, presentation and popularization of the Bulgarian cultural and historic heritage (CHH)
2. Development of information systems and platforms with geolocation of the CHH for research, preservation and popularization
3. Creation of R&D and educational programs related to the Bulgarian CHH.

The expected results of the research program are as follows:

- New ways to present the scientific results of the humanitarian studies, including the use of "Linked Open Data"
- Products that improve the awareness of the general public regarding the CHH and the national identity (University of Sofia, 2019)

The Technical University of Sofia participates in the research program by developing digital instruments for research, presentation and popularization of the CHH and an information system with geolocation of the CHH. During the first year of the research program, the focus was on the development of the core of the software needed for the project. For the second and third year, integrating existing datasets created by the other partners and improving the end-user experience becomes the priority.

Three of the datasets created by a participant in this project contain geographical objects, mostly settlements, covering the area of Bulgaria. As all datasets refer to mostly the same settlements, it would be useful to merge them in a single dataset and look for further insights. This paper presents a way to merge these, and possibly other similar datasets, using a heuristic approach.

## ANALYSIS OF THE SPATIAL DATASETS

The three datasets to be merged come from three different sources:

- Dataset 1 was created by toponym map-mining of the Russian “triverstova” map in scale 1:126 000
- Dataset 2 was created by toponym map-mining of the Austrial “Generalkarte” in scale 1:200 000
- Dataset 3 is a manually created dataset based on Ottoman documents and the Bulgarian “EKATTE” nomenclature.

## Data Format and Volume

All datasets were provided as OpenXML spreadsheets, each containing a sheet where each row represents one object. Only dataset 3 had two more sheets – one with a description of the columns and another with descriptions of administrative units – both sheets were not needed for the merge.

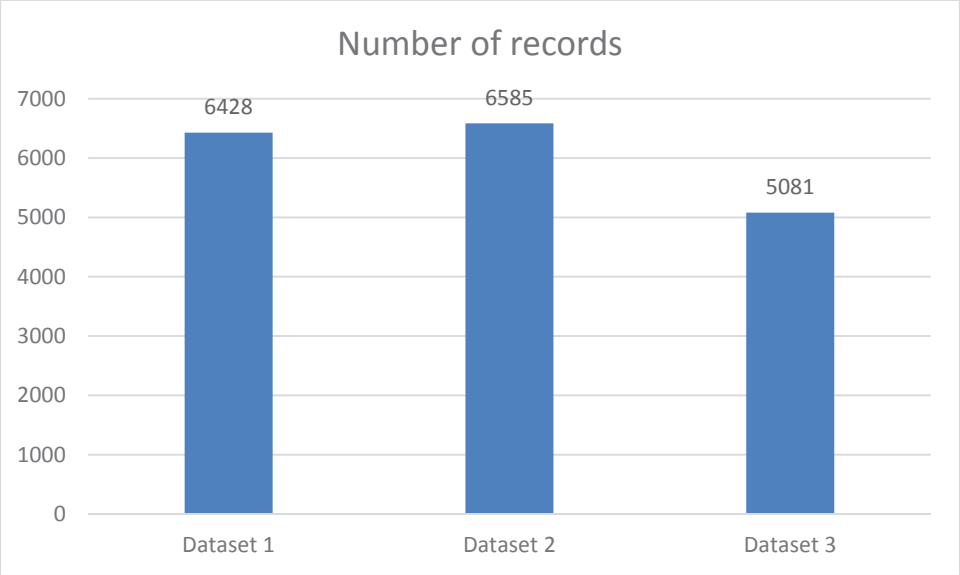


**FIGURE 1**. Number of records in each dataset.

The number of records in each dataset was different, as seen in Fig. 1. This suggests objects in one dataset may be missing from one or more of the other datasets. Also, the number of objects is too large for a manual merge, so some form of automation is needed to perform this task.

Next, we turn to the attribute data in each dataset.

Dataset 1 has the following attributes:

1. Geographical latitude
2. Geographical longitude
3. Name in Russian
4. Description
5. Alternative name
6. Type

The attributes of Dataset 2 are:

1. Geographical latitude
2. Geographical longitude
3. Name
4. Description
5. Alternative name
6. Type
7. Map sheet number

The attributes of Dataset 3 are:

1. Geographical latitude
2. Geographical longitude
3. Name in Ottoman Arabic
4. Name transliteration in modern Turkish
5. Alternative name (in modern Turkish)
6. Name in Bulgarian
7. Suffix in English (ex. “v.” for village)
8. Name in English
9. EKATTE
10. District code
11. Municipality code
12. Township code
13. Municipality name
14. Comment

### Data to Consider

When merging two datasets into one, we need something that is present in both of them, a “link” that allows us to match a record in one datasets with record(s) in the other.

Looking at the attributes of the three datasets, we notice that all of them have geographical coordinates. This attribute is a good potential candidate for a link, the problem being that the coordinates do not match *exactly* – randomly handpicked objects show a variation around 200-300 meters between different datasets.

Another candidate attribute is the name or the alternative name of the object. All datasets have it, so why don’t we match them by name? The problem here is that the same object has a different name in every dataset. Dataset 3 has the most variants for the name – Ottoman Arabic, Turkish, Bulgarian and English – and yet does not have Russian (as in Dataset 1) and even the Turkish names in Dataset 2 and Dataset 3 often don’t match, as they also appear to be from different time periods.

Another potential attribute is the “type” field. In theory, we could eliminate false positives for matches if their types are different. Looking into this approach, we found that nearly 92% of all objects had the same type – settlement. Unfortunately, this greatly reduces the potential benefit of this approach.

As a conclusion, the geographical coordinates seemed to be the beast candidate for a link field.

## A HEURISTIC FOR MERGING SPATIAL DATASETS

A simple algorithm that can merge two datasets works as follows:

1. Take a point from one dataset and find nearby points in the other dataset
2. Pick the best candidate from the other dataset
3. Merge the two records into one by combining their attribute data.

Step 1 implies some criteria for “nearby” points. When the data itself is not very precise, these criteria do not need to be precise either. We could express “nearness” as fractions of a decimal degree in both latitude and longitude.

Step 2 is a resolution step: given two or more records to choose from, we need to choose one that is an actual match. One solution is to minimize the number of candidates, so we have many records with only one candidate that can be merged automatically.

### Implementation of the Heuristic

We implemented the heuristic above using an SQL-driven approach. We first added a row number (ID) attribute to each dataset for easier identification of rows in the different datasets.

The next step is to move the data to an SQL database where we can use queries to implement the algorithm. We chose PostgreSQL because it is a popular open-source database. Spreadsheets cannot be imported into PostgreSQL directly but an open-source tool, pgfutter (lukasmartinelli, 2018), can import CSV files into a PostgreSQL database.

Therefore, we first converted the spreadsheets to CSV. We then used pgfutter to create three tables for the three datasets and imported the data there.

We then performed a series of SQL queries to select an appropriate search distance. We will discuss those queries in the next section. For now, we focus on merging the datasets. We made a two-phase merge, where we first merged Dataset 1 and Dataset 2, exported the dataset using pgAdmin's "Download as CSV" function, performed the manual merges where necessary, imported the result as a new dataset, and then merged Dataset 3 with the merged dataset. The SQL to merge two datasets is presented in Fig. 2.

```
SELECT d1.*, d2.*
FROM d1
LEFT JOIN d2
ON ABS(d2."Easting" - d1."Easting") < 0.003 AND ABS(d2."Northing" - d1."Northing") < 0.003
UNION ALL
SELECT d1.*, d2.*
FROM d1
RIGHT JOIN d2
ON ABS(d2."Easting" - d1."Easting") < 0.003 AND ABS(d2."Northing" - d1."Northing") < 0.003
WHERE d1."ID1" IS NULL
```

**FIGURE 2**. An SQL query for merging two datasets

This query selects the columns from both datasets, effectively merging them. Both datasets may contain rows that do not have a match in the other dataset, so two subqueries are joined together using UNION ALL. Matches from the second query that are also in the first are eliminated using the WHERE d1."ID1" IS NULL clause. If there are multiple nearby matches for a record, we would get this record multiple times in the result set, merged with every candidate match. We need to resolve such conflicts manually, which we will discuss shortly.

## Selecting an Appropriate Search Distance

The query in Fig.2 uses 0.003 decimal degrees as the maximum offset between two "nearby" points but how did we choose this number? It is a balancing act between two effects:

1. If the distance is too small we may fail to merge two identical places
2. If the distance is too big we will have too many candidates to manually pick from

The limiting factor for us the second point – we need to select a distance that gives us only a few cases with more than one candidate. Effectively, an appropriate search distance is the biggest distance that gives reasonably few candidates that need manual intervention for the entire dataset. But then comes the question "How many candidates do we have that need manual intervention"? We can use SQL to answer this question for a specific search distance, as shown in Fig. 3.

```
SELECT Merged."ID", COUNT(Merged."ID1") FROM
(SELECT d1.*, d2.*
FROM d1
LEFT JOIN d2
ON ABS(d2."Easting" - d1."Easting") < 0.003 AND ABS(d2."Northing" - d1."Northing") < 0.003
UNION ALL
SELECT d1.*, d2.*
FROM d1
RIGHT JOIN d2
ON ABS(d2."Easting" - d1."Easting") < 0.003 AND ABS(d2."Northing" - d1."Northing") < 0.003
WHERE d1."ID1" IS NULL) as Merged
GROUP BY Merged."ID"
HAVING COUNT(Merged."ID1") > 1
```

**FIGURE 3**. An SQL query that gives the number of matches for each record in the first dataset that has more than one candidate match in the second dataset.

We use a query on top of our merge query to count the number of rows for each row in our initial dataset to get how many candidates we have for each row. This query is useful because it also gives the IDs of the rows we need to merge manually. A COUNT(*) on top of this query gives us the total number of rows that need to be merged manually. We can now find a reasonable search distance by running this query multiple times, giving a different search distance and observing the number of results. Practically, this means starting with a very low value and a very high value, and then performing a binary search in that search space for a good value.

## RESULTS

For the first merge, a distance value of 0.003 decimal degrees, which is approx. 333 meters (Approximate Metric Equivalents for Degrees, Minutes, and Seconds, 2019), yielded 10 rown that needed manual merge. For the second merge, the same value yielded 16 rows. The three datasets with 18094 rows were merged into one dataset with 8802 rows, thus 9292 rows were merged successfully using the heuristic approach we presented.

The merged dataset was imported as a layer in the interactive web-based system with geolocation created for this project. The results are shown in Fig. 4.

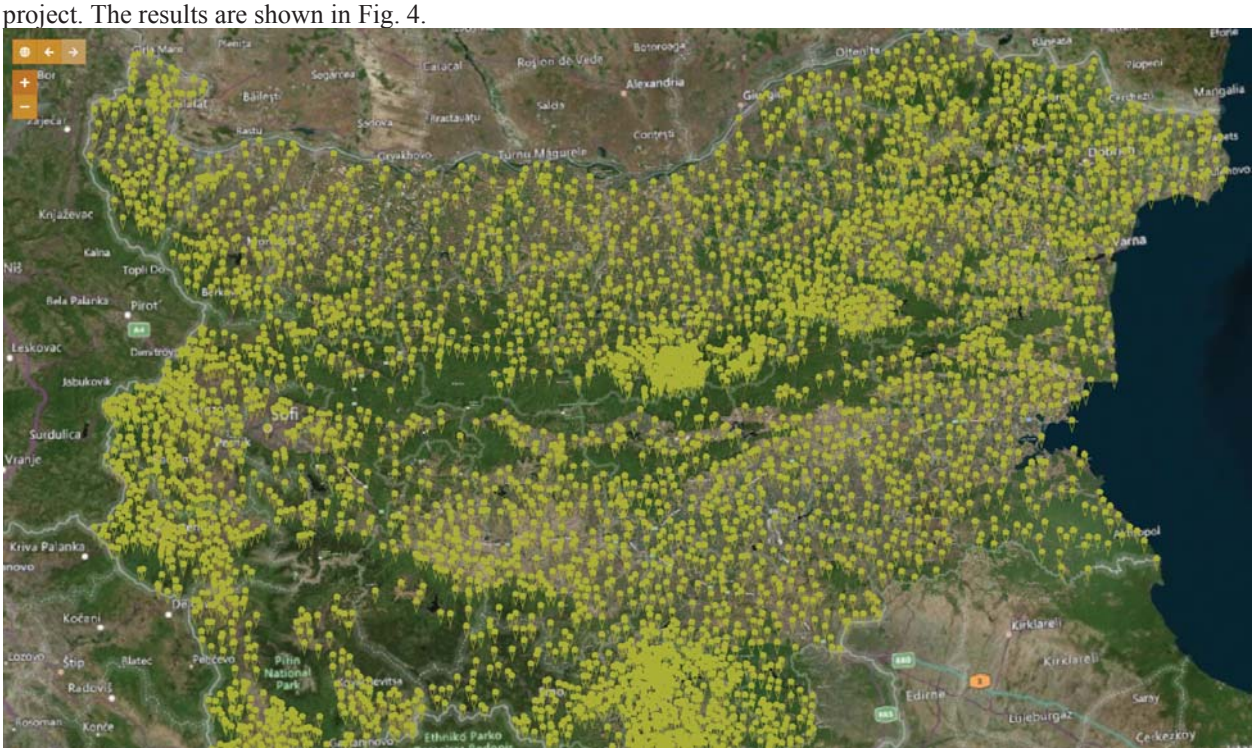


**FIGURE 4**. The merged dataset, as a layer in an interactive web-based system with geolocation.

In areas with higher density of objects, such as the villages north of Kazanlak, and around Kardzali, the results were visually inspected for false positives on merges or missed merges, but none were found.

In conclusion, we presented a simple yet effective method for merging spatial datasets when the objects are sparsely distributed relative to each other.

## ACKNOWLEDGMENTS

The work (paper, book, monograph, conference) was supported/(partially supported) by the Bulgarian Ministry of Education and Science under Cultural Heritage, National Memory and Social Development National Research Program approved by DCM No 577 of 17 August 2018.

## REFERENCES


1. *Approximate Metric Equivalents for Degrees, Minutes, and Seconds*. (2019, August 5). Retrieved from https://www.usna.edu/Users/oceano/pguth/md_help/html/approx_equivalents.htm
2. Council of Ministers of The Republic of Bulgaria. (2018, 08 17). DCM No 577. Sofia, Bulgaria.
3. lukasmartinelli. (2018). *pgfutter.* Retrieved from pgfutter: https://github.com/lukasmartinelli/pgfutter
4. University of Sofia. (2019). *Natinal Scientific Programme „Cultural and Historical Heritage, National Memory and Social Development"*. Retrieved from KINNPOR: https://kinnpor.uni-sofia.bg/programme