

This is an Accepted Manuscript of a book chapter published by IGI Global in the “Spatial Planning in the Big Data Revolution” book (<https://doi.org/10.4018/978-1-5225-7927-4>) on 15.03.2019, available online: <https://doi.org/10.4018/978-1-5225-7927-4.ch002>

# Modelling and Assessing Spatial Big Data

## Use Cases of the OpenStreetMap Full–History Dump

Alexey Noskov  
*Heidelberg University, Germany*

A. Yair Grinberger  
*Heidelberg University, Germany*

Nikolaos Papapesios  
*University College London, UK*

Adam Rousell  
*Heidelberg University, Germany*

Rafael Troilo  
*Heidelberg University, Germany*

Alexander Zipf  
*Heidelberg University, Germany*

### ABSTRACT

*Many methods for intrinsic quality assessment of spatial data are based on the OpenStreetMap full-history dump. Typically, the high-level analysis is conducted; few approaches take into account the low-level properties of data files. In this work, a low-level data-type analysis is introduced. It offers a novel framework for the overview of big data files and assessment of full-history data provenance (lineage). Developed tools generate tables and charts, which facilitate the comparison and analysis of datasets. Also, resulting data helped to develop a universal data model for optimal storing of OpenStreetMap full-history data in the form of a relational database. Databases for several pilot sites were evaluated by two use cases. First, a number of intrinsic data quality indicators and related metrics were implemented. Second, a framework for the inventory of spatial distribution of massive data uploads is discussed. Both use cases confirm the effectiveness of the proposed data-type analysis and derived relational data model.*

Keywords: Intrinsic Data Quality, Spatial Distribution, Users’ Activity, Contributors’ Activity, Tcl, Parallel Processing, London, Turin, Venice, Tel-Aviv, Gaza Strip

## **INTRODUCTION**

Nowadays, effective processing of big data files is a significant challenge. Standard and well-known big data files are images and video files. In addition to them, a tremendous amount of information is registered in the form of log files. Data-centered services often record all contributions and modifications of datasets. In contrast to private data hidden from the public, open data services, like Wikipedia and OpenStreetMap (OSM), usually provide the history of all users' contributions in the form of full-history dumps (FHD). It is popular to provide full-history information of open data projects in either compressed XML format or Google's Protocolbuffer Binary Format (PBF).

Classically, big data files are converted to an indexed relational database for further processing. Recently, various big-data specific solutions have been introduced. Typically, such solutions are based on multi-core cloud solutions. The MapReduce concept is usually implemented for the development of software for multi-core based processing of big data files. Many approaches to big log data processing have been introduced in the last years. Apache Hadoop and Apache Spark are popular platforms for MapReduce-based novel solutions.

In this work, OSM FHD data are considered. Even though the article is focused on OSM, prospectively, proposed solutions can be extended to other sources of full-history data (e.g., Wikipedia). All introduced approaches are developed in an open-source manner. Hence, the developed solutions can be quickly adapted and improved for other projects. Several prepared tools are assembled as a part of Integrated Geographic Information Systems Tool Kit (IGIS.TK), which can be described as IDE for GIS projects. Currently, the parallelization is achieved by concurrent processing of FHD's clips. In the future, it will be modified by using threading libraries for optimal utilization of available CPUs.

In this work, a data-type model for universal analysis of full-history data is introduced. The model provides an overview of FHD and insight into the data provenance. The model is designed for the low-level data-type based analysis of full-history data. It allows comparing different clipped FHDs and observing dynamics and specific of users' contributions. Resulting statistics are presented in table and chart views. Table and charts are available as interactive HTML files. Moreover, using the prepared statistics, a novel relational data model for OSM FHD has been developed. Databases for several pilot sites have been generated according to the relational model.

Two use cases are based on the prepared databases. In the frame of the first use case, various intrinsic data quality indicators and related metrics were calculated. The resulting data and charts allow users to compare the quality of the examined FHD datasets. In the frame of the second use case, the spatial distribution of massive data uploads is investigated.

## **BACKGROUND**

Nowadays, processing of big data files is often focused on log data. Kreps et al. (2011) discussed this problem and proposed their solutions. They noticed that a vast amount of log data produced by large internet services. For instance, every day, China Mobile collects up to 8TB of phone call

records, while Facebook harvests about 6TB of data related to user activity. For this, distributed log aggregators (Facebook's Scribe, Yahoo's Data Highway, and Cloudera's Flume) are delivered by various companies.

Such solutions can be described as traditional enterprise messaging systems. They play a role as an event bus for processing asynchronous data flows. For instance, IBM Websphere MQ allows applications to insert messages into multiple queues atomically. Some systems do not enable batching numerous messages into a single request; this raises performance issues. To resolve this, solutions like Facebook's Scribe aggregate log separately and, then, periodically dump them to HDFS. Moreover, various similar solutions are offered by Cloudera, Yahoo, and LinkedIn. All these applications can be described as "messaging systems." Modern messaging systems support asynchronous distributed logging and processing.

Messaging systems described earlier collect log data. These data are used for various reasons, like monitoring of web services, exploring security and performance issues, debugging, etc. It requires applications for effective processing of big log files. Such tools were discussed by Breier and Branišová (2015). Often, these solutions are considered as data mining solutions or, in other words, software, which uses statistical and machine learning algorithms for disclosing standard and non-standard properties described by the processed information. It is very popular to use Apache Hadoop as a processing environment and MapReduce as a programming model. MapReduce (Xhafa et al., 2015) is a popular parallel-programming model for massive Big Data processing. Two phases are implemented by the model: map and reduce. Each stage has key-value pairs as input and output. Xhafa et al. (2015) explored the performance and usability of the Apache Hadoop system run on 64 nodes. They utilized big log data collected by Virtual Campus web applications, which manage about 46GB per day. A MapReduce application was implemented for this.

Breier and Branišová (2015) described another MapReduce implementation utilized for the anomaly detection from log files; it is based on data mining techniques. The application supports several logging types: security (reveal potential breaches, malicious programs, information thefts and to check the state of security controls), access, operational, compliance. They used three main detection methods for monitoring malicious activities: scanning, activity monitoring and integrity check.

Another type of log data is applications' debugging information. Ghoshal and Plale (2013) carried out research contributing to this topic. In the frame of the mentioned work, they have defined data provenance as the lineage of a data set, capturing transformations or derivations applied. This term is intensively utilized in the present article. Ghoshal and Plale considered the following log levels: trace, debug, info, warn, error and fatal.

Moreover, they distinguished two main classes of provenance: process provenance and data provenance. Quality of provenance data was examined in the context of several application cases: completeness, redundancy, timestamp, and error and failure. A relatively similar use case was covered by research conducted by Hao et al. (2016). An application was implemented for

processing of game/simulation-based assessment log information. In contrast to the mentioned applications, a solution based on Python/Scipy/Pandas was developed.

Chaiken et al. (2008) discussed the disadvantages of the MapReduce model, where a map function performs grouping and a reduce function performs aggregation. The parallelism is achieved by splitting data into small parts and consequent parallel independent processing of these parts. The limitations of this model are as follows: the necessity of the map phase implementation (even for unnatural causes, like projection or selection) and the difficulty of the development of a complex application, which requires multiple map-reduce stages. In order to resolve considered issues, they introduced SCOPE. SCOPE is a SQL-like declarative programming language designed by Microsoft for the massive parallel computing.

In addition to normal log data processing, mentioned solutions can be effectively utilized for performing analysis of massive full-history dumps (FHDs). FHDs are provided by projects collecting volunteered information. Wikipedia and OpenStreetMap are two recognizable projects. The both provide FHDs as a compressed large XML file or a binary protobuf file. Various projects utilize Wikipedia's FHD. For instance, the BDpedia project (Lehmann et al., 2015) implements services based on the Wikipedia dump. Structured multilingual knowledge from Wikipedia is extracted and made freely available on the web using the semantic web and linked data technologies.

Among works strongly relied on Wikipedia FHD, one can mention a paper on Wikipedia's edit history efficiency assessment (Ferschke et al., 2011). The article introduces an application, which facilitates the processing of full-history data by reducing the amount of data. It was proposed to apply the described solution by Natural Language Processing (NLP) algorithms. In addition to this, Pasternack and Roth (2010) suggested a framework for the trustworthiness assessment of information provided by Wikipedia. Like the previous one, it relies on Wikipedia's FHD. One can conclude that the performance of both solutions can be dramatically improved by utilization of the parallelization using systems like Apache Hadoop. For instance, the first application (Ferschke et al., 2011) is based on the Java Wikipedia library (JWPL). Normal processing of Wikipedia's FHD can take several days; a proposed by the documentation solutions (through the parallelization) allows reducing processing time to 1 to 3 days, which is still quite long.

Like Wikipedia, the OpenStreetMap (OSM) project offers FHD. OSM's FHD is considered in various research works overviewed in (Auer et al., 2018). The authors of the mentioned work propose "the OHSOME software platform that applies big data technology to the OSM full history data." It is implemented in and designed for the Java programming language. Two case studies for disaster management were discussed. In the present work, an alternative software platform is proposed; it is biased to the lower-level data processing.

## **SOURCE DATA PREPARATION**

The planet service of OpenStreetMap provides FHD files in either XML.BZ2 or Protocol Buffers data format. In the frame of the present work, only data in XML.BZ2 format are processed. The current size of FHD is 64G4; the uncompressed size of the archive is about 1TB.

Most OSM tools allow processing uncompressed data. For clipping FHD in required areas, OSM-history-splitter (MaZderMind, 2017) has been used. Data have been clipped in 5 pilot sites: San Donà di Piave (SD), Turin (TR), Southwark (SW), Heidelberg (HD), Israel (IS). SD is a town and part of the Metropolitan City of Venice. Both, SD and TR reside in Italy. SW is a district of Central London, UK. HD is a town in Baden-Württemberg, Germany. The IS pilot site covers the whole area of Israel and neighbor areas around the country.

Further processing has been conducted using tools designed within the IGIS.TK project (Noskov, 2018). The following tools are involved. First, `c/osh2sql.tcl` is designed for converting OSM raw data for SQLite relational database. Moreover, it can collect statistics useful for the low-level analysis of OSM full-history data. Second, `q/inventosmd3.tcl` is developed for visualization the collected by `c/osh2sql.tcl` statistics. Third, `q/introsmd3.tcl` implements a framework for the intrinsic quality assessment of OSM full-history data; it relies on some existing well-known approaches. Both tools, i.e., `q/inventosmd3` and `q/introsmd3.tcl`, provide a series of D3/Charts.JS graphs and tables and store results of the visualization in HTML files. It allows presenting results of the inventory as a part of Geo-Spatial Data Repository (Noskov & Zipf, 2018). All mentioned tools are open source; anyone can download them from the online repository.

`c/osh2sql.tcl` generates SQLite database comprising OSM full-history data and line/tag statistics. All tables of SQLite resulting databases are indexed; thus, these data are suitable for the high-performance data analysis using regular SQL queries. Users may utilize any programming language and environment for implementing queries because SQLite is a very wide-spread and flexible library. IGIS.TK can be used as either SQL/Tcl code editor or command line console for preparing and executing for SQL queries. Line/tag statistics are saved in plain text files; these files are processed by `q/inventosmd3` to visualize the results.

The `planet-latest.osm.bz2` file was downloaded in June 2018. Using OSM-History-Splitter (MaZderMind, 2017), the FHD file was clipped according to the areas of the pilot sites. The downloading and clipping require about 8 hours. Then, the following command was carried out to generate SQLite databases and stats files.

```
tclsh ~/prs/igistk/c/osh2sql.tcl -i "| bunzip2 -c
/home/fudeb/projectsdata/dq/southwark.osh.bz2" -s southwark.osh.sqlite -t
tagstats.txt -l linestats.txt
```

The “-i” parameter sets the input files. Two file formats can be used with the parameter. First, uncompressed \*.osh XML files can be set. In order to utilize compressed \*.osh.bz2 files, the input string starting with “| bunzip2 -c ...” should be used. In this case, Tcl programming language, instead of opening a file directly, opens a socket which reads the output of bunzip2 Unix (MacOS, GNU/Linux) command. In Windows, an analog to bunzip2 command should be exploited. The “-s” parameter defines a path to an output database file. The “-t” and “-l” parameters set paths to outputs tag and line statistics.

Prepared raw statistics is not suitable for human analysis. For this, a tool for visualization of stats files was developed. The following command demonstrates the utilization of this tool.

```
tclsh ~/prs/igistk/q/inventosmd3.tcl -l "sandona_linestats.txt,  
turin_linestats.txt,southwark_linestats.txt,hd_linestats.txt,israel_linestats  
.txt" -t "sandona_tagstats.txt, turin_tagstats.txt, southwark_tagstats.txt,  
hd_tagstats.txt,israel_tagstats.txt" -n "San Donà di Piave, Turin, Southwark,  
Heidelberg, Israel" -o /tmp/fhdp_invn.html
```

The “-l” and “-t” parameters (optional, but at least one of them has to be set) define input line and tag (correspondingly) stats files separated by commas. The “-n” parameter provides names of pilot sites; it is used in output tables and charts. The “-o” parameter sets a path to an output HTML file comprising tables and D3 chars suitable for human analysis.

The prepared SQLite databases were utilized to calculate some measures and charts required intrinsic quality assessment of OSM full-history data. The following commands are carried out for this.

```
tclsh ~/prs/igistk/q/introsmd3.tcl -i "San Donà di Piave, sandona.osh.sqlite:  
Turin,turin.osh.sqlite:Southwark,southwark.osh.sqlite:Heidelberg,hd.osh.sqlit  
e" -n sandona.poly,turin.poly,southwark.poly,hd.poly -o /tmp/fhdp_intr.html
```

The “-i” parameter defines names of pilot sites and paths to databases separated by commas; colons separate names of pilot sites. “-o” sets output HTML file.

OSM-History-Splitter utilizes \*.poly files for establishing input areas for clipping. The q/introsmd3.tcl tool can digest these files for the normalization of results. In order to increase the results’ usability, data should be normalized. It is possible to normalize the results using the area or boundary length of a pilot site. It was decided that both metrics are not applicable. In the case of Israel, a significant part of the clipping polygon is covered by the Mediterranean Sea. Using the area for the normalization will senselessly decrease the output metrics. The length of a boundary is also not an option, because for pilot sites with detailed curvy boundary the length will be much bigger than for rectangular or circular areas; thus, such normalization is also senseless. In order to normalize the data and eliminate the mentioned problems, the length of a boundary of the convex hull of a pilot site’s polygon is used. Paths to \*.poly files for the normalization are provided by “-n” parameter; it is optional.

## **DATA-TYPE ANALYSIS OF FULL-HISTORY DUMP**

### **Concept**

The present article offers a data-type analysis approach. All programming languages provide various data-type systems. Most languages distinguish Boolean, Integer, Float and String data types. One can call it “basic data types.” Boolean is True or False value. Alternatively, it can be 1/0 or yes/no. The string consists of characters; characters can be either ASCII or non-ASCII.

Moreover, various classes can be distinguished: digits, letters, spaces, lower- and upper-case, punctuations, printable, etc. In this work, a data-type system and character classes offered by the Tcl programming language are utilized to analyze the OSM full-history data. It allows defining imperfections in source datasets and discloses the provenance of full-history information.

## Line Statistics

Line statistics is calculated for raw lines of XML files regardless of the document object model. Thus, if an object occupies several lines, each line will be assessed separately. This analysis allows users to detect imperfections of data and obtain a general overview of data. The number of lines and number of characters are two primary and trivial statistical metrics that are calculated. The ratio of the number of characters per line can be used as a quality indicator. It shows the density of FHD files. The number of lines is utilized for normalization of other parameters.

Further, a data-type model for calculation of line statistics is provided. Firstly, a line is trimmed to calculate how many space characters are resided at the starting (*sblank*) and ending of a line (*fblank*). Starting spaces (spaces or tabs) are used for indents. Ending spaces usually mark imperfection of data writing processes. Starting and ending spaces are excluded from further analysis. Next, the rest of a line is split to attribute (*attrs*) and non-attribute (*noattrs*) classes. “Attributes” means strings in double quotes. *noattrs* is all string outside double quotes. In the following three-line sample of XML data “36”, “true”, “2”, “way”, “268” and “inner” are *attrs*; all other strings are *noattrs*.

```
<relation id="36" visible="true" version="2">
  <member type="way" ref="268" role="inner"/>
</relation>
```

In the XML listing, one can mention the special characters: less, more, slash, equal and space. Less and less-slash start an XML instruction; more character closes an instruction. The equal sign joins Key-value attributes; spaces separate words. First, the XML starting instruction characters’ frequency is calculated: less followed by a word (*stags*), less-slash (*ctags*) and less not followed by a word (*less*). *less* detects errors. Second, the ending instruction frequency is calculated: *slashmore* and *more*. Then, number equal signs are calculated. All recorded characters are excluded from further consideration. Next, space characters are reported. Spaces are split into two classes: one space character followed by non-space character (*mblanks1*) and several neighbor space characters (*mblanksmore*). Moreover, the frequency of equal signs is reported.

Now, the rest *noattrs* characters are evaluated one by one. Five classes are distinguished: ASCII digits (*noattrs09*), ASCII lower-case letters (*noattrsaz*), ASCII upper-case letters (*noattrsAZ*), the rest ASCII characters (*noattrsASCII*), and other characters (*noattrsany*).

Finally, *attrs* strings are evaluated. First of all, double quote frequency is calculated with the following elimination from the further processing. The rest characters are split into eleven classes. The first class is for spaces (*attrsblanks*). Three ASCII classes follow it: digits (*attrs09*), lower (*attrsaz*) and upper-case (*attrsAZ*) letters. Next, non-ASCII letters in lower (*attrslow*, e.g. Cyrillic letters а-я) and upper-case (*atrsup*, e.g., Cyrillic letters А-Я) are reported. Then, other letters without upper or lower case are reported (*atrsalpha*, e.g., Hebrew letters א-ת). Furthermore, non-ASCII digits are detected (*atrsdigit*, e.g., Thai digit seven ๗). And finally,

punctuation (*atrspunct*) and printable (*atrsgaph*) characters are defined. All rest characters belong to the class *atrsany*. It discloses imperfections of an FHD file.

## Tag Statistics

“Tag statistics” is a kind of historical name. The first prototype of the tool dealt with tags only, thus “tag” work was deeply hardcoded in the source code. Currently, the tool works with properties of all XML elements (nodes, ways, relations and members), thus, “tag” meaning could be interpreted as a description of XML elements. Moreover, some consider XML elements (especially HTML element, like, <p>, <form>, <ul>) as “tags”. More accurate meaning of “tag statistics” in the present work expressed as statistics regarding the values of attributes organized according to XML elements (or tags). Each XML elements has attributes encoded as key-value pair joined by the equal sign; values are enclosed in double quotes. For instance, in the attribute *type="way"*, *type* is key of the attribute and *way* is value. Thus, tag statistics is a collection of data-type information regarding XML attribute values for obtaining and visualization of data provenance information.

Document Object Model (DOM) of an XML data is a hierarchical tree of elements. <osm> is the main element of \*.osh documents. It contains <node>, <way> and <relation> elements. Each of them may comprise <tag> elements. Topological connections are established by either <nd> elements of <way> or <member> elements of <relation>. <nd> points to existing <node> element; <member> points to existing <relation> or <way> elements. Each <node>, <way> and <relation> element has *timestamp* attribute. That allows deriving provenance information. Line stats represent recorded data outside of attribute values quite carefully in detail. This information is well established and generated mainly by software, while value information have relatively free format. Significant part of value information comes from users. Thus, it is quite important to implement deep analysis of value information.

Further, a model for deep analysis of value information is presented. Statistics is assigned to entities named according to the following scheme: *maintag::tag::key::datatype*. As mentioned, OSH XML file have <osm> root element; it is ignored by the model. <node>, <way> and <relation> are *maintag* elements. If attribute values of *maintag* elements are considered, tag is equal to *maintag* (e.g., *node::node::\** or *way::way::\**). Otherwise, *maintag* represents a root element of the considered element (e.g., *node::tag::\**, *relation::member::\**, *way::nd::\**). It is followed by a key of an attribute (e.g., *relation::member::role::\**, *node::node::lon::\**, *way::tag::v::\**).

An entity name ends by data type number. The used list of numbered data type is as follows: (0) BLANK, (1) UNDEFINED, (2) NOASCIILIST, (3) NOASCIANY, (4) NOASCIIALPHA, (5) ASCIILIST, (6) ALPHA, (7) ASCIANY, (8) DOUBLE, (9) INTEGER, (10) BOOL. Thus, *way::way::visible::10* means key “visible” of way element (<way> has no root elements, its name is duplicated in the entity) has the Boolean type. Notice that Boolean data can be recognized from the following case-insensitive forms: 0/1, false/true, no/yes and n/y. Hence, building height property equals “1” could be wrongly recognized as Boolean. Unfortunately, because of some minor imperfections in the tool implementation, some values are recognized as

the UNDEFINED class. This class can be considered as an unspecified abstract string. Few of the following entity instances belong to the UNDEFINED: `relation::member::type`, `relation::relation::user`, `relation::relation::changeset`, `relation::relation::uid`, `relation::relation::id`, `node::node::id`, `relation::relation::timestamp` and `relation::relation::visible`. It should be fixed in the future. This bug does not affect the current results and can be considered as a minor shortage of the implementation.

It should be mentioned that, in contrast to line stats, for tag stats values are considered as a whole value's string without splitting to separate characters. In the data classes, BLANK means an empty value or a value containing only spaces. UNDEFINED class was described earlier. NOASCII LIST implies that string includes no-ASCII characters and spaces separating non-space characters. Such can be processed as Tcl list with length more than 1. In Tcl, lists are string separated by space characters. NOASCII ANY is a value containing non-ASCII characters without spaces, excluding ending and starting spaces, which can be trimmed. NOASCII ALPHA is a value comprising non-ASCII letters. ASCII LIST represents values containing ASCII only characters without spaces. ALPHA is a value comprising only letters. ASCII ANY describes all others cases with ASCII-only characters that do not belong to any other ASCII classes described earlier. DOUBLE, INTEGER and BOOL are for Float, Integer, and Boolean data types.

Moreover, entities of tag statistics files have time range prefix. Statistics is aggregated for every 3 months from the starting of the OSM project (i.e., 01-10-2004 00:00:00). Time interval is presented a starting point of an interval (or tick). Number of a tick is used as prefix. Numbered list of ticks is as follows: (0) 01/10/04, (1) 01/01/05, (2) 01/04/05, (3) 01/07/05, (4) 01/10/05, (5) 01/01/06, (6) 01/04/06, (7) 01/07/06, (8) 01/10/06, (9) 01/01/07, (10) 01/04/07, (11) 01/07/07, (12) 01/10/07, (13) 01/01/08, (14) 01/04/08, (15) 01/07/08, (16) 01/10/08, (17) 01/01/09, (18) 01/04/09, (19) 01/07/09, (20) 01/10/09, (21) 01/01/10, (22) 01/04/10, (23) 01/07/10, (24) 01/10/10, (25) 01/01/11, (26) 01/04/11, (27) 01/07/11, (28) 01/10/11, (29) 01/01/12, (30) 01/04/12, (31) 01/07/12, (32) 01/10/12, (33) 01/01/13, (34) 01/04/13, (35) 01/07/13, (36) 01/10/13, (37) 01/01/14, (38) 01/04/14, (39) 01/07/14, (40) 01/10/14, (41) 01/01/15, (42) 01/04/15, (43) 01/07/15, (44) 01/10/15, (45) 01/01/16, (46) 01/04/16, (47) 01/07/16, (48) 01/10/16, (49) 01/01/17, (50) 01/04/17, (51) 01/07/17, (52) 01/10/17, (53) 01/01/18, (54) 01/04/18, (55) 01/07/18, (56) 01/09/18. Hence, `32::node::tag::*` means that the `<tag>` of the `<node>` belongs to a time interval from 01/10/12 to 01/01/13.

For each entity, four elements are recorded: summarized length of strings, number of evaluated values, minimal length of string or minimal value (of the value belongs for integer or float numbers) and maximal length of string or maximal value (of the value belongs for integer or float numbers). If a data type is Boolean, then, instead of the definition of minimal and maximal values, False and True values are recorded correspondingly.

In the following bulleted list, some entity samples from the raw tag stats file of the Southwark pilot site generated by the tool and their descriptions are provided.

- **39::relation::member::role::5 {381 26 9 23}**: Time interval 39 (i.e., 01/07/14 - 01/10/14), `<member>` of root element `<relation>` has values of the "role" key, which belong to type

5 (ASCIILIST). Total length of all strings is 381 characters; 26 values were recorded; shortest value has 9 characters; longest – 23.

- **19::way::tag::k::6 {62340 10717 2 12}**: Time interval 19 (i.e., 01/07/09- 01/10/09), <tag> of root element <way> has values of “k” keys, which belong to type 6 (ALPHA). Total length of all strings is 62340 characters; 10717 values were recorded; shortest value has 2 characters; longest – 12.
- **25::node::tag::v::7 {16984 1705 1 116}**: Time interval 26 (i.e., 01/01/11 - 01/04/11), <tag> of root element <node> has values of “v” keys, which belong to type 7 (ASCIIANY). Total length of all strings is 16984 characters; 1705 values were recorded; shortest value has 1 character; longest – 116.
- **33::node::node::lon::8 {274807 27483 -0.2157702 0.0369058}**: Time interval 33 (i.e., 01/01/13 - 01/04/13), <node> element has values of “lon” (longitude) keys, which belong to type 8 (DOUBLE). Total length of all strings is 274807 characters; 27483 values were recorded; minimal value is -0.2157702; maximal is 0.0369058.
- **33::node::node::lon::8 {274807 27483 -0.2157702 0.0369058}**: Time interval 33 (i.e., 01/01/13 - 01/04/13), <node> element has values of “lon” (longitude) keys, which belong to type 8 (DOUBLE). Total length of all strings is 274807 characters; 27483 values were recorded; minimal value is -0.2157702; maximal is 0.0369058.
- **42::way::way::version::9 {5973 5462 1 49}**: Time interval 42 (i.e., 01/01/13 - 01/04/13), <way> element has values of “version” keys, which belong to type 9 (INTEGER). Total length of all strings is 5973 characters; 5462 values were recorded; minimal version of the “way” is 1; maximal is 49.
- **12::way::way::visible::10 {48545 12007 517 11490}**: Time interval 12 (i.e., 01/10/07 - 01/01/08), <way> element has values of “visible” keys, which belong to type 10 (BOOL). Total length of all strings is 48545 characters; 12007 values were recorded; 517 ways were deleted (the “visible” key’s values are “false”); 11490 ways of the Southwark data are active (the “visible” key’s values are “true”).

## Results

The presented data-type models are implemented by `c/osh2sql.tcl` tool of the IGIS.TK project. The results are saved in two text files with raw records. The tool `q/introsmd3.tcl` has been designed for the visualization of these text files in a human-readable form. Further, a general overview of the visualized results is provided for line and tag statistics. As mentioned, line statistics give an overall review of considered OSM data dumps, while tags statistics provide a detailed insight into the low-level evolution of data.

### *Line Statistics*

In Table 1, a general overview of FHD files is provided. Columns provide data according to different pilot sites. Rows represent various data types. Further, various data-type classes are considered from top to bottom of the table. Mostly, character number is provided. In parenthesis, the number of characters divided by the number of lines of an FHD file (see the first row from the top). Normalized numbers are provided in an exponential format.

The pilot sites are ordered ascendingly from left to right column of the resulting table by the number of lines and characters as follows: SD, TR, SW, HD and IS. It is very important to normalize the resulting metrics because the variation of pilot sites' areas is high. Thus, it makes no sense to compare raw values. It has been decided to use the number of lines of a considered FHD for the normalization. Hence, values in parenthesis right below corresponding raw integers are normalized. Further, the normalized results are considered according to their classes from top to bottom of the table.

*sblank* indicates the number of nested elements because child XML elements are written with an increasing starting indent. The value is growing gradually from SD to IS from left to right. Thus, IS has more nested elements and deeper XML trees in comparison to other FHDs. *fblank* class indicates imperfections of input data files because OSM XML tags should be written without any ending spaces. All the values equal "0"; thus, no problems have been recognized by the *fblank* class indicator.

IS, HD, and SW comprise relatively similar number *stags* per line. SD and TR FHDs have much less tag starting characters (i.e., the less sign followed by a letter). *stags* represents the number of XML elements. In contrast to it, *ctags* distinguish starting instructions of a closing tag (i.e., the less sign followed by the slash). This class represents the number of elements having child elements. Hence, IS has much more nested tags than all others pilot sites. It confirms the finding derived from the analysis of the *sblank* class. The difference of the meaning of the results based on *sblank* and *ctags* is that, *ctags* is showing the number of elements having classes, while *stags* reflects the number of child elements and depth of an XML tree. The *less* class helps to detect imperfection of formatting tags (i.e., the number of cases when the less sign is not followed by a letter). For the considered FHDs, no problems were detected by this class. *slashmore* indicates the number of one-line XML elements without child elements. IS and HD have the maximal number of such elements per line; they followed by SW, TR, and SD descendingly. As *ctags*, *more* shows the number of XML tags with child elements. Notice that *more* equals *ctags* minus one.

*blanks1* and *blanksmore* examine space characters in FHDs. *blanksmore* disclose imperfections of XML files (outside of attribute values, number of neighbor space characters should not be more than 1, excluding indents). Such shortcomings were not disclosed in the considered data files. *blanks1* indicates the amount of XML entities separated by spaces. It is decreasing from left to right. Then, the possible rest variation of values is examined in the non-attribute-value strings. Among *nonatrs\** classes, the only *nonatrsaz* has non-zero values. That indicates well formed OSH files. *dquotes* class reflects the number of attributes. *dquotes* raw values must be even integers. Again, the normalized values of the table are gradually decreased from left to right.

Further, attribute values (*atrs\**) are considered. One crucial issue should be mentioned. Non-ASCII values are encoded as XML symbol entities. For instance, a string "באר" in Hebrew and encoded into string "&#x5D1;&#x5D0;&#x5E8;" in an OSH XML data file. Hexadecimal digits starting are used. Hence, the quantity of the mentioned characters is dramatically increased in FHDs containing non-ASCII symbols. Thus, one can observe anomalies of the *atrs\** classes ratio

Table 1. Line statistics

Attribute name	San Donà di Piave	Turin	Southwark	Heidelberg	Israel
lines	6,697,526	7,819,818	25,963,886	39,117,310	71,403,578
chars	336,660,937 (4.715E+00)	413,949,061 (5.797E+00)	1,352,982,944 (1.895E+01)	1,955,812,558 (2.739E+01)	4,778,721,665 (6.693E+01)
sblank	26,093,138 (3.654E-01)	28,859,126 (4.042E-01)	99,119,404 (1.388E+00)	152,097,436 (2.130E+00)	240,219,384 (3.364E+00)
fblank	0	0	0	0	0
atrs	135,671,359 (1.900E+00)	198,373,021 (2.778E+00)	603,786,723 (8.456E+00)	810,088,473 (1.135E+01)	2,587,887,640 (3.624E+01)
noatrs	174,896,440 (2.449E+00)	186,716,914 (2.615E+00)	650,076,817 (9.104E+00)	993,626,649 (1.392E+01)	1,950,614,641 (2.732E+01)
stags	43,161,665 (6.045E-01)	40,056,175 (5.610E-01)	153,452,855 (2.149E+00)	246,679,654 (3.455E+00)	266,523,320 (3.733E+00)
ctags	90,636 (1.269E-03)	820,100 (1.149E-02)	1,253,524 (1.756E-02)	1,029,206 (1.441E-02)	5,701,186 (7.984E-02)
less	0	0	0	0	0
slashmore	13,213,782 (1.851E-01)	13,999,438 (1.961E-01)	49,420,726 (6.921E-01)	76,176,210 (1.067E+00)	131,404,786 (1.840E+00)
more	90,635 (1.269E-03)	820,099 (1.149E-02)	1,253,523 (1.756E-02)	1,029,205 (1.441E-02)	5,701,185 (7.984E-02)
mblanks1	20,309,981 (2.844E-01)	22,320,326 (3.126E-01)	76,628,332 (1.073E+00)	115,951,671 (1.624E+00)	232,704,576 (3.259E+00)
mblanksmore	0	0	0	0	0
noatrs09	0	0	0	0	0
noatrsaz	77,719,760 (1.088E+00)	86,380,450 (1.210E+00)	291,439,525 (4.082E+00)	436,809,032 (6.117E+00)	1,075,875,012 (1.507E+01)
noatrsAZ	0	0	0	0	0
noatrsascii	0	0	0	0	0
noatrsany	0	0	0	0	0
dquotes	40,619,962 (5.689E-01)	44,640,652 (6.252E-01)	153,256,664 (2.146E+00)	231,903,342 (3.248E+00)	465,409,152 (6.518E+00)
atrsblanks	114,529 (1.604E-03)	961,327 (1.346E-02)	3,222,440 (4.513E-02)	1,000,328 (1.401E-02)	9,040,857 (1.266E-01)
atrs09	66,195,216 (9.271E-01)	90,172,104 (1.263E+00)	261,391,671 (3.661E+00)	379,470,399 (5.314E+00)	1,467,798,691 (2.056E+01)
atrsaz	25,411,468 (3.559E-01)	51,360,874 (7.193E-01)	161,844,023 (2.267E+00)	177,569,944 (2.487E+00)	360,127,476 (5.044E+00)
atrsAZ	949,441 (1.330E-02)	4,370,843 (6.121E-02)	9,630,328 (1.349E-01)	6,811,893 (9.540E-02)	97,285,361 (1.362E+00)
atrslow	0	0	0	0	0
atrsup	0	0	0	0	0
atrsalpha	0	0	0	0	0
atrsdigit	0	0	0	0	0
atrspunct	2,376,719 (3.329E-02)	6,864,688 (9.614E-02)	14,433,469 (2.021E-01)	13,303,219 (1.863E-01)	188,188,756 (2.636E+00)
atrsgraph	4,024 (5.636E-05)	2,533 (3.547E-05)	8,033 (1.125E-04)	28,658 (4.014E-04)	37,341 (5.230E-04)
atrsany	0	0	95 (1.330E-06)	690 (9.663E-06)	6 (8.403E-08)



defected characters outside have not been detected outside of attribute value. Thus, they do not have a negative impact on other XML elements.

### *Tag Statistics*

Tag statistics allows provenance assessment of OSM full-history data. As mentioned, data are aggregated for every three months from the beginning of the OSM project. The results of the aggregation are provided according to the XML tag tree, then, an XML key followed by the examining value, and, finally, the data type of a value. Because it is impossible to present highly-granulated data in the frame of the current article, the resulting data is aggregated according to the second level of the granularity (i.e., the parent tag and child tag).

In Figure 3, the results are demonstrated. Graphs on the left hand represent the summarized number of characters of values; graphs on the right hand illustrates the number of values. The statistics regarding the following entities is provided: `node::node`, `node::tag`, `way::way`, `way::nd`, `way::tag`, `relation::relation`, `relation::member` and `relation::tag`. Repeating tag name means that a tag does not have a parent tag.

The most recognizable entity in the graphs is `relation::member`; it is followed by `node::node`, `way::node`, `node::tag`, `relation::tag` and `way::tag`. In all pilot sites, excluding IS, `relation::member` occupied a dominating space. In SD and Turing, the number of such objects was much higher at the beginning of the OSM project 2008 – 2012. After 2012, the number of commits has been decreased significantly. Nowadays, the activity of OSM contributors is quite low in both pilot sites. In SW, the activity is very high and increasing from at the beginning of mapping until today. In HD, the amount of contributions is high as well, but it is slightly decreased since 2013.

In addition to the `relation::member`, in SD FHD, one can notice recognizable contributions of `node::node` and `way::node` (the letter is mainly identifiable in the left chart). In 2017, charts of TR show a significant peak of commits of `node::node` and `node::tag`. Hence, one can expect that the quality of POI data has been increased significantly after this. In SW charts, the activity related to `node::node`, `node::tag` and `way::nd` is presented from the beginning until now. In 2015, contributions of `relation::tag` are distinguishable in the left chart. In HD, `node::node` and `way::node` contributions are observable (especially from 2011 to 2015), but it plays much less role in comparison to SW data.

IS has a particular pattern which is highly distinguishable from the other pilot sites. The pattern mainly consists of `node::node`, `way::nd` and `node::tag`. Therefore, the contribution primarily consists of providing new node data, creating `<way>`s linked to `<node>` and committing `node::tag` data. Both chart types confirm it. Moreover, in IS datasets, recognizable peaks are illustrated. It can be related to political and military issues which happened at a time of emerging of these peaks.

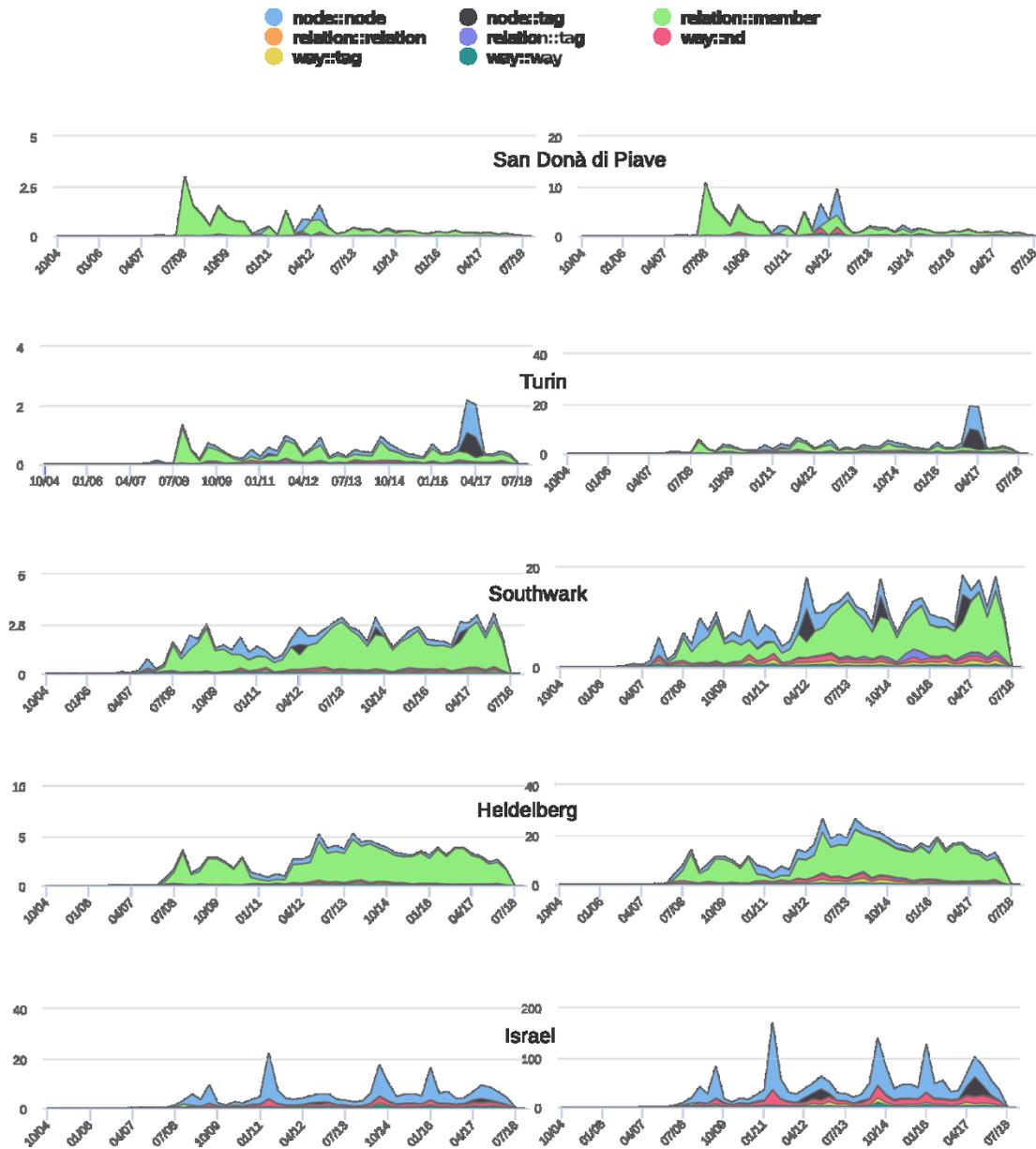


Figure 3. Tag statistics results

## OSM FULL-HISTORY DATA MODEL

The conducted data-type analysis allows us to prepare a rectified data model for management OSM data as an indexed relational database. Usually, OSM is managed by PostGIS; it is aimed to standard non-full-history data (i.e., \*.osm\* and not \*.osh\* source data files). We propose to use an SQLite database for the OSM full-history management. A novel rectified relational data model is proposed in the present article; it is based on the results discussed in the previous section.

The proposed data model is based on the analysis of the considered line and tag statistics. The full list of examined data-types of OSM FHDs is as follows.

```
node::node::changeset::9, node::node::id::9, node::node::lat::8,
node::node::lon::8, node::node::timestamp::9, node::node::uid::9,
node::node::user::5, node::node::user::6, node::node::user::7,
node::node::version::9, node::node::visible::10, node::tag::k::0,
node::tag::k::5, node::tag::k::6, node::tag::k::7, node::tag::v::0,
node::tag::v::5, node::tag::v::6, node::tag::v::7, relation::member::ref::9,
relation::member::role::0, relation::member::role::5,
relation::member::role::6, relation::member::role::7,
relation::member::type::1, relation::relation::changeset::1,
relation::relation::id::1, relation::relation::timestamp::1,
relation::relation::uid::1, relation::relation::user::1,
relation::relation::version::1, relation::relation::visible::1,
relation::tag::k::0, relation::tag::k::5, relation::tag::k::6,
relation::tag::k::7, relation::tag::v::0, relation::tag::v::5,
relation::tag::v::6, relation::tag::v::7, way::nd::ref::9, way::tag::k::0,
way::tag::k::5, way::tag::k::6, way::tag::k::7, way::tag::v::0,
way::tag::v::5, way::tag::v::6, way::tag::v::7, way::way::changeset::9,
way::way::id::9, way::way::timestamp::9, way::way::uid::9, way::way::user::5,
way::way::user::6, way::way::user::7, way::way::version::9,
way::way::visible::10
```

A novel data model for the relational database has been designed using the presented tag entities consisting of tag names, an attribute name, and a data type index. The novel data model is not based or related to a popular OSM data model which is used for PostGIS databases (OpenStreetMap, 2016).

In Figure 4, the proposed model is presented. *Elements* is a main table of the data model. *id* is a unique identifier of OSM history object (i.e., every version of an OSM object has a unique identifier). *xmlid* is not a unique value; it is taken from an XML's file. Various versions share the same identifier. *version* is a version of an element. *type* can be "0" (node), "1" (way) or "2" (relation). *uid* is a user identifier. *visible* can be either "true" or "false"; it indicates if an object is either active or disabled (removed). *timestamp* is an object creation time. *changeset* is a changeset's identifier.

*uid* provides references to the *id* field of the *Users* table. *Users* names are stored in the *name* field of the *Users* table; it prevents from the text values duplication. As usernames, OSM tags key and values strings are stored in separate tables in *txt* fields – *Keys* and *Vals*, correspondingly. The *Tags* table establishes pairs of tags' keys and values. The table provides references to *Key* and *Vals* through the *key* and *val* fields.

Geometry objects are formed by the topology definition using the *Elidxy* and *Relations* tables. Nodes are points which are bases for all other geometries. *Elidxy* establishes one-to-many relationships with *XYs* containing *x* and *y* coordinates. In addition to nodes, ways and relation

geometric objects can be defined by the *Relations* table. The *Relrols* and *Roles* tables specify *Relations*; recursive processes construct ways and relations.

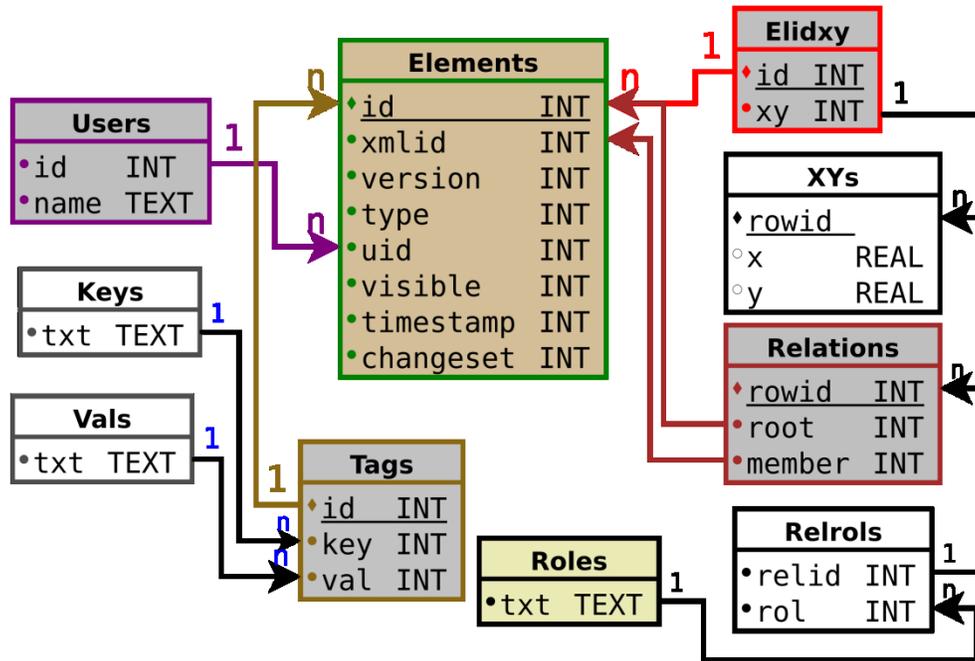


Figure 4. The data model of OSM full-history data.

The *c/osh2sql.tcl* tool of IGIS.TK converts an FHD file to an SQLite database file according to the presented data model. In the next two sections, two use cases of the prepared database file utilization. First, a framework for the intrinsic quality assessment of OSM full-history data is provided. Second, an approach to spatial distribution of the massive data uploads is presented as the second use case.

### USE CASE 1: INVENTORY AND QUALITY ASSESSMENT OF OPENSTREETMAP DATA – INSTINSIC APPROACH

The *q/introsmd3.tcl* tool of IGIS.TK generates an HTML file comprising various charts useful for intrinsic and comparable assessment of OSM full-history data. Charts are generated for provided pilot sites. In Figure 5, calculated charts for SD, TR, SW, HD and IS are presented. On the left-hand side, raw data are presented; on the right hand, normalized data are presented. As discussed, data are normalized using the length of a boundary of a polygon’s convex hull. Calculated lengths utilized for the normalization are as follows: SD - 41 km, TR - 44 km, SW - 42 km and HD - 64 km.

Data and correspondent charts *a1*, *a2*, *b1*, and *b2* are utilized by various approaches for intrinsic and comparable quality assessment of VGI data. For instance, Girres and Touya (2010) used

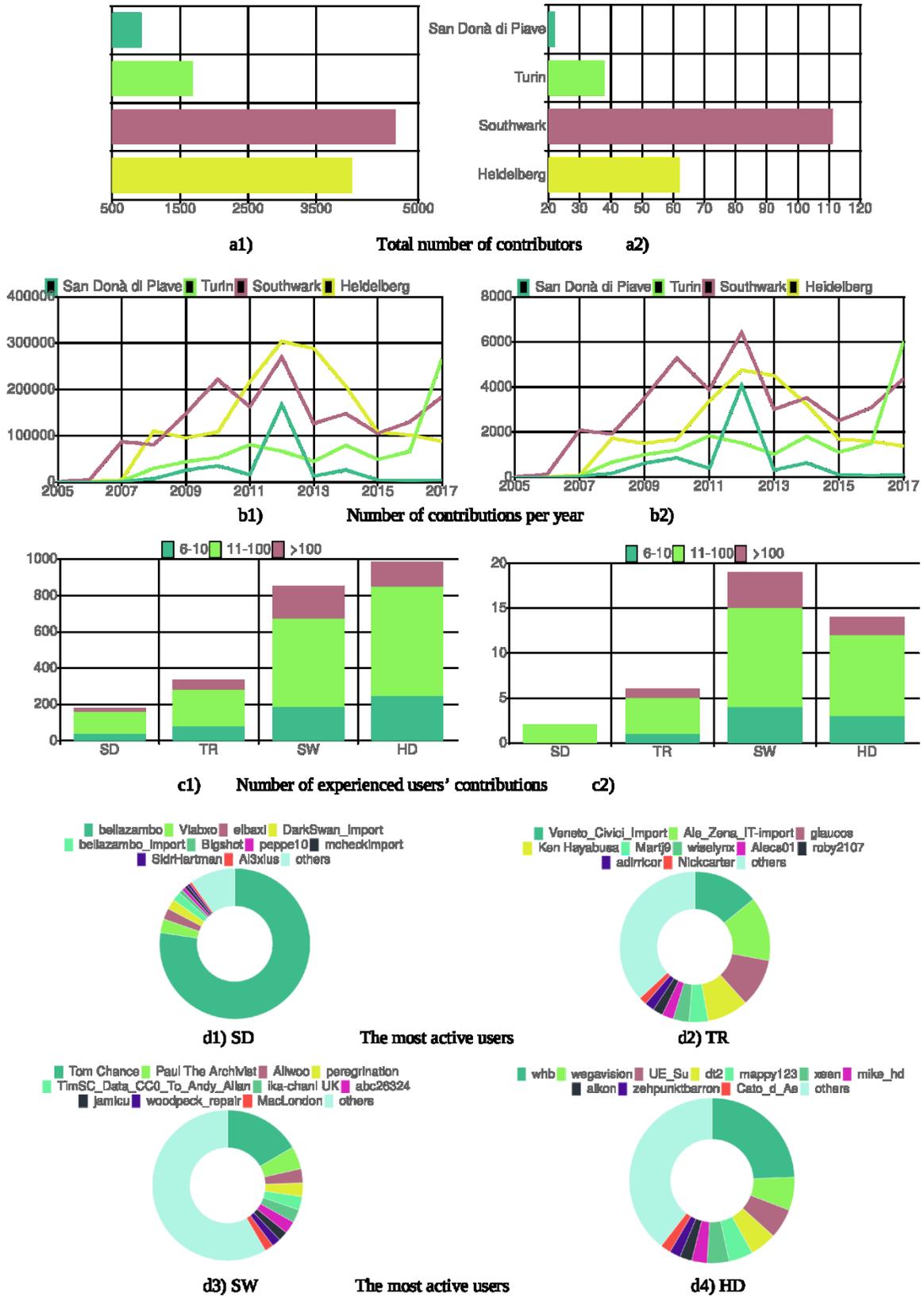


Figure 5. Carts for intrinsic and comparable quality assessment of OSM data

such data for the lineage-based quality assessment of OSM data. *a1* and *a2* represent the number of contributors registered by FHDs and its normalized version. Notice the impact of the normalization on increasing the differences in the resulting values. The normalized quantity of contributors in SW is much bigger in comparison to others, especially the pilot sites in Italy. *c1* and *c2* show the dynamics of the number of contributions (number of elements with a correspondent timestamp) and its normalized values. One can mention that, after the normalization, SW is followed by HD.

Further, charts related to a trustworthiness aspect of the data quality are discussed (Kessler et al., 2013). Versions, the number of users' commits and the overall users' distributions are affected by the trustworthiness of OSM FHD data. *c1* and *c2* represent the number of users with more than five contributions with a version more than 5. Most of the users have 11 to 100 contributions. As in the previous charts, these charts confirm that SW FHD provides the highest quality of the dataset. It is followed by HD, TR, and SD, descendingly. *d1-d4* illustrate the distribution of contributions among various users. Variegated charts and charts with a bigger ratio of the "others" category indicate higher quality datasets.

It should be mentioned that, according to the presented charts in Figure 5, SD FHD provides the lowest quality dataset. Charts of TR indicate higher quality. With a significant gap, the data quality is increased from TR to HD. SW delivers the highest quality dataset significantly distinguished from the other pilot sites; it is showed by the all normalized charts of Figure 5. This fact is confirmed by collected line and tag statistics provided by Table 1, Figure 1 and Figure 3.

## **USE CASE 2: THE SPATIAL DISTRIBUTION OF MASSIVE DATA UPLOADS IN TEL AVIV-YAFFO AND THE GAZA STRIP**

To explore the utility of the suggested data structure, we utilize it for studying the spatial distribution of OSM contributions in the city of Tel Aviv-Yaffo (TLV) in Israel and the Gaza strip (GZS). Some interesting patterns in these areas were noted before, where data in GZS is created mostly through external interventions which lead to massive contributions over a short period (Bittner, 2017) while TLV is characterized by a more gradual increase in dataset size, except for one event of a massive data import (Grinberger, 2018). Such massive events were found to affect data quality in terms of richness, the frequency of updates, and community structures (Grinberger, 2018), yet the spatial dimension of these dynamics have yet to be studied.

This use case adds to this by focusing on three massive data events (Table 2) – one for TLV in which an official addresses database which was made publicly available by governmental agencies was imported into OSM on December 2013 via an effort coordinated within the local community of OSMappers (yrtimiD, 2012); two for GZS, the first of which organized by a NGO which hired local residents to map the road network in the strip during 2009 (i.e. GZS-2009; JumpStart Mapping, 2009) and the second carried as part of a Humanitarian OSM Team (HOT) project during the summer of 2014 and the months following it, focusing on remotely mapping buildings within GZS using a high-resolution aerial image of the area (i.e. GZS-2014; OpenStreetMap Wiki Contributors, 2014). As noted above, these different dynamics and their

relations to access the mapped area introduce different effects to data quality. Accordingly, they are expected to affect also the spatial coverage of contributions.

Table 2. Characteristics of data events.

<b>Event</b>	<b>GZS-2009</b>	<b>TLV-2012</b>	<b>GZS-2014</b>
<b>Time Period</b>	21-22/09/2009	22/12/2012	01/08/2014-30/11/2014
<b>Organizer</b>	JumpStart International	Local OSM Community	Humanitarian OSM Team (HOT)
<b>Focus</b>	Roads	Addresses	Buildings
<b>Method of contribution</b>	Land survey	Data import	Remote mapping
<b>Bounding Box Coordinates(lat/lon – WN,ES - WGS84)</b>	34.2, 31.2 34.6, 31.6	34.72, 32.03 34.85, 32.14	34.2, 31.2 34.6, 31.6
<b># new nodes</b>	81,307	53,130	952,335
<b># new tagged nodes (% of total)</b>	2,541 (3.12%)	53,130 (100.00%)	50,324 (5.28%)

To better understand the coverage patterns of the data produced during each event (i.e. TLV-2012, GZS-2009, and GZS-2014), the framework suggested in this chapter was utilized to identify the nodes created during each event and to distinguish between nodes enriched with semantic information (i.e. ‘tags’) and nodes with no such information. For this, a series of simple SQL queries were written, used to join the elements table with coordinates and tags data and to filter contributions by time and location (see Table 2). The resulting dataset for each event was aggregated into a grid covering the study area with a spatial resolution of 250 square meters. For each cell, the total number of newly created nodes and the number of new tagged nodes was recorded.

The TLV-2012 event, which was based on a systematically collected authoritative dataset, can serve as a reference for the GZS events which are collected in a different manner. For instance, the spatial distribution of new nodes (Figure 6a) mirrors to a large extent the urban structure, with the historic cores of Tel Aviv and Yaffo densely covered and the relatively newer and wealthier neighborhoods of the north presenting lower densities. The same is true for the GZS events, yet to a lesser extent, especially in the case of the 2009 event where data densities do not obey to municipal boundaries as strictly as in the 2014 event (Figures 6b and 6d). While this can be explained by the focus of each event on a different class of entities (Table 2), the differences in the coverage of semantic information (Figures 6c and 6e) require a different explanation. The picture these present is opposite to the overall picture – in the GZS-2009 event mostly urban centers are covered while in GZS-2014 the pattern is less random.

When these patterns are quantified by counting the number of new nodes within and outside official municipal boundaries (Table 3), these contrasting trends become even more evident – although only a third of the 2009 contributions are made within urban areas (and almost 20% less than in GZS-2014), a much greater share of these are tagged with semantic information in relation 2014, and a slightly larger share of all tagged nodes are concentrated within urban areas. The explanation for this may lie in how ancillary knowledge is gathered and semantic

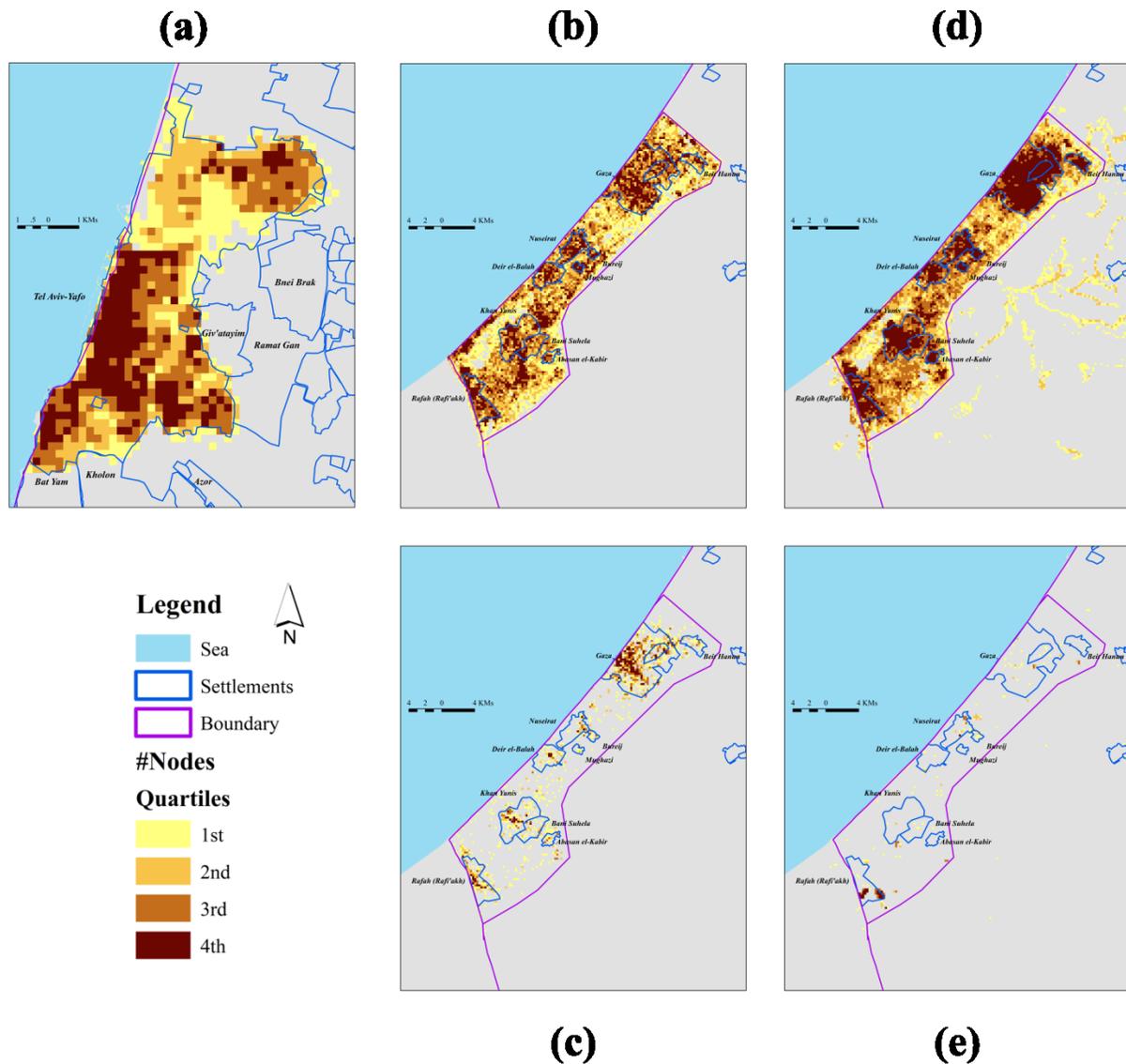


Figure 6. Density of new entities and new tagged entities by event and case study: (a) TLV-2012, new nodes; (b) GZS-2009, new nodes; (c) GZS-2009, new tagged nodes; (d) GZS-2014, new nodes; (e) GZS-2014, new tagged nodes.

information is produced in the two cases. The 2014 mappers had to rely only on visual assessments of the aerial image, meaning tags were only created when the image (or the existing data) provided relevant ‘clues’, leading a somewhat random pattern. The local residents mapping during the 2009 event, however, relied on their experience and local knowledge to identify what is ‘important’ and worthy of integration into the dataset. Hence, it is not surprising urban centers which take significant roles within the everyday lives of individuals are better represented. This analysis of data coverage, utilizing the suggested data structure, thus uncovers spatial patterns related to coverage, semantic information, and mapping dynamics that have not received full attention within the existing literature to this date. Understanding these via data structures that

facilitate high-resolution mapping of data production dynamics can thus greatly contribute to the assessment of data quality.

*Table 3. Distribution of nodes, by event, tags, and urban areas*

<b>Measure</b>	<b>GZS-2009</b>	<b>TLV-2012</b>	<b>GZS-2014</b>
<b>% nodes within urban areas</b>	33.80%	100.00%	49.61%
<b>- of these: % tagged</b>	5.57%	100.00%	0.68%
<b>% tagged nodes within urban areas</b>	64.26%	100.00%	58.38%

## **FUTURE RESEARCH DIRECTIONS**

As shown, the presented data-type model provides a novel type of line and tag statistics. In the present article, only part of generated data is considered. The implemented model generates highly granulated data, which should be presented and discussed in the future work. The introduced processes for data-type identification should be slightly refined. In order to demonstrate the advantages of the solutions, more chart types need to be utilized. A more in-depth analysis of the resulting data and charts should be conducted in the future.

As mentioned, the current implementations of the discussed tools contain minor bugs; it will be fixed in the next releases of IGIS.TK. Currently, only command line tools are implemented. In the next stage, their GUI wrappers will be prepared. IGIS.TK provides the functionality for the rapid development of such GUI wrappers and manages them as parts of IGIS.TK's IDE (main GUI programming environment comprising source-code editor, command-line console, map widgets, and SQLite database manager). Moreover, binary packages of IGIS for the delivery and quick installation of the software should be prepared for Unix (MacOS, GNU/Linux, BSD), Windows and Android.

A framework for the intrinsic quality assessment of OSM full-history data will be significantly extended and improved. Currently, few quality indicators and related measures are implemented. The list should be considerably expanded. Consequently, a broader analysis of the resulting data and charts should be conducted. Since the proposed relational data modes of OSM full-history data is universal, more used cases can be considered in the future.

## **CONCLUSIONS**

The present work introduces the novel data-type model for the inventory of OSM full-history data. The model is implemented as the tool of the IGIS.TK open-source software. Any user may evaluate the proposed solutions using other parts of OSM FHD. Furthermore, because the software is released as an open-source project, anyone can improve the code and contribute modifications to the project.

The data-type model generates the line and tag statistics. The line statistics provides a general overview of examined FHDs. Much information can be extracted from the line statistics. The normalization (by the number of lines) allows users to compare FHDs covering non-similar (by size) areas. Apart from that, the line statistics help to detect imperfections of FHD. It can be a result of incorrect either clipping of OSM FHD or preparation OSM FHD covering the whole planet. At least one problem has been detected and discussed in the result section. The tag statistics is useful for the data provenance analysis because the resulting data are aggregated by every three months (the time interval can be modified). The tag statistics shows the low-level dynamics of the OSM XML object model and distinct data types of values of XML tags attributes. Attribute values stores most information contributed by volunteers.

The introduced tools generate HTML5 charts. Main charts were discussed in this work. Several inferences have been disclosed in the charts and tables. The proposed data-model for the line statistics allows detecting imperfection in examined data. In FHD provided by the OSM planet, problem characters indicating possible faults in the process of full-history data dumping have been found. The tag statistics distinguished three lineage types of OSM FHDs. First, the pilot sites in Italy are not gradually developed; a significant part of the data is contributed by bulk imports. Very low contributors' activity follows short periods of massive contributions; that indicates possible problems with data quality in these Italian pilot sites. Second, the Southwark and Heidelberg pilot sites comprised information contributed gradually and intensively from the beginning of the OSM project till now. In contrast to Heidelberg, Southwark data are still contributed to the growing trend. Third, the Israel pilot site is distinguished by both type of committed data and the fact that contribution peaks are significantly related to the political and military events in the Middle East.

In addition to the mentioned inferences, the data-type model resulting data utilized for developing an optimal universal relational data model for storing and managing OSM full-history data. Each FHD was converted to the correspondent SQLite indexed relational file database according to the presented data model. These databases were utilized in the two discussed use cases. The use cases have confirmed the findings concluded from the data-type assessment in higher-level. It affirms that the introduced data-type analysis offers researchers a valuable set of tools for investigating full-history data.

## ACKNOWLEDGMENT

This work has been funded by the European Union's Horizon 2020 research and innovation programme under the grant agreement n. 693514 ("WeGovNow"). The article reflects only the authors' view, and the European Commission is not responsible for any use that may be made of the information it contains.

## REFERENCES

### Journal articles:

- Auer, Michael & Eckle, Melanie & Fendrich, Sascha & Griesbaum, Luisa & Kowatsch, Fabian & Marx, Sabrina & Raifer, Martin & Schott, Moritz & Troilo, Rafael & Zipf, Alexander. (2018). Towards Using the Potential of OpenStreetMap History for Disaster Activation Monitoring. *Proceedings of the 15th ISCRAM Conference* – Rochester, NY, USA May 2018. Kees Boersma and Brian Tomaszewski, eds. pp. 317-325.
- Bittner, C. (2017) OpenStreetMap in Israel and Palestine – ‘game changer’ or reproducer of contested geographies? *Political Geography*, 57, 34-48.
- Breier, J., & Branišová, J. (2015). Anomaly detection from log files using data mining techniques. *Information Science and Applications*, Springer, Berlin, Heidelberg, pp. 449-457
- Girres, J., & Touya, G., (2010). Quality assessment of the French OpenStreetMap dataset. *Transactions in GIS*, 14(4), pp.435-459 <https://doi.org/10.1111/j.1467-9671.2010.01203.x>
- Hao, J., Smith, L., Mislevy, R., von Davier, A., & Bauer, M. (2016). Taming Log Files From Game/Simulation-Based Assessments: Data Models and Data Analysis Tools. *ETS Research Report Series*, 2016(1), pp. 1-17.
- Kessler, C., Theodore, R., & De Groot, A., (2013) Trust as a Proxy Measure for the Quality of Volunteered Geographic Information in the Case of OpenStreetMap, *Geographic Information Science at the Heart of Europe*, pp. 21–37.

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., ... & Bizer, C. (2015). DBpedia—a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*, 6(2), pp. 167-195 <https://doi.org/10.3233/SW-140134>.

Sawyer, S., & Tapia, A. (2005). The sociotechnical nature of mobile computing work: Evidence from a study of policing in the United States. *International Journal of Technology and Human Interaction*, 1(3), 1-14.

### **Published proceedings:**

Deci, E. L., & Ryan, R. M. (1991). A motivational approach to self: Integration in personality. In *Proceedings of Nebraska Symposium on Motivation* (vol. 38, pp. 237-288). Lincoln, NE: University of Nebraska Press.

Chaiken, R., Jenkins, B., Larson, P. Å., Ramsey, B., Shakib, D., Weaver, S., & Zhou, J. (2008). SCOPE: easy and efficient parallel processing of massive data sets. *Proceedings of the VLDB Endowment*, 1(2), pp. 1265-1276.

Ferschke, O., Zesch, T., & Gurevych, I. (2011). Wikipedia revision toolkit: efficiently accessing Wikipedia's edit history. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Systems Demonstrations, Association for Computational Linguistics*, pp. 97-102.

Ghoshal, D., & Plale, B. (2013). Provenance from log files: a BigData problem. In *Proceedings of the Joint EDBT/ICDT 2013 Workshops*, ACM, pp. 290-297.

Kreps, J., Narkhede, N., & Rao, J. (2011). Kafka: A distributed messaging system for log processing. In *Proceedings of the NetDB*, pp. 1-7.

Noskov, A. & Zipf, A. (2018), Backend and Frontend Strategies for Deployment of WebGIS Services. Proc. SPIE 10773, *Sixth International Conference on Remote Sensing and Geoinformation of the Environment* (RSCy2018). <https://doi.org/10.1117/12.2322831>

Noskov, A. (2018). Open Source Tools for Coastal Dynamics Monitoring, Proc. SPIE 10773, *Sixth International Conference on Remote Sensing and Geoinformation of the Environment* (RSCy2018), Vol 107731C. <https://doi.org/10.1117/12.2326277>

Pasternack, J., & Roth, D. (2010, August). Knowing what to believe (when you already know something). In *Proceedings of the 23rd International Conference on Computational Linguistics, Association for Computational Linguistics*, pp. 877-885.

Khafa, F., Garcia, D., Ramirez, D., & Caballé, S. (2015). Performance Evaluation of a MapReduce Hadoop-Based Implementation for Processing Large Virtual Campus Log Files. In *5th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)*, IEEE, pp. 200-206.

### **A presented paper:**

Grinberger, A. Y. (2018). *Identifying the effects of mobility domains on VGI: Towards an analytical approach*. Paper presented at the VGI-ALIVE workshop, AGILE18, Lund, Sweden. Available from: <http://www.cs.nuim.ie/~pmooney/vgi-alive2018/papers/1.2.doc>

### **Website:**

Noskov, A (2018). Integrated Geographic Information System - Tool Kit. Retrieved from <http://igis.tk> or <http://igis.n-kov.com>

MaZderMind (2017). *OSM History Splitter*. Retrieved from <https://github.com/MaZderMind/osm-history-splitter>

VandenBos, G., Knapp, S., & Doe, J. (2001). *Role of reference elements in the selection of resources by psychology undergraduates*. Retrieved from <http://jbr.org/articles.html>

OpenStreetMap contributors (2016)  
[https://wiki.openstreetmap.org/w/images/5/58/OSM\\_DB\\_Schema\\_2016-12-13.svg](https://wiki.openstreetmap.org/w/images/5/58/OSM_DB_Schema_2016-12-13.svg)

JumpStart Mapping. (2009) FreeMap Gaza Launched. [Online] Available from: <http://jumpstart-mapping.blogspot.de/2009/04/freemap-gaza-launched.html> [Accessed 29th March 2018].

OpenStreetMap Wiki Contributors. (2018) 2014 Gaza Strip. [Online] Available from: [https://wiki.openstreetmap.org/wiki/2014\\_Gaza\\_Strip](https://wiki.openstreetmap.org/wiki/2014_Gaza_Strip) [Accessed 29th March 2018]

yrtimiD. (2012) New gov data source. [Online] Available from: <https://forum.openstreetmap.org/viewtopic.php?id=19240> [Accessed 29th March 2018].

## KEY TERMS AND DEFINITIONS

**D3 (or D3.js):** is a JavaScript library generating interactive data visualizations, mainly charts, according to SVG, HTML5, and CSS web standards.

**GIS:** Geographic Information Systems.

**GUI:** Graphic User Interface.

**Data Mining:** is the process of discovering patterns in large datasets involving methods at the intersection of machine learning, statistics, and database systems.

**Hadoop (Apache Hadoop):** An operating system developed in the frame of the Apache project; the system allows distributed calculation among various detached virtual and physical servers called nodes.

**HDFS:** A distributed file system developed in the frame of the Apache Hadoop system.

**IDE:** Integrated Development Environment.

**Log Data (Log Files):** Data files containing the information registered consequently from oldest to newest (e.g., debug information provided by an application, users' requests registered by a web server, etc.).

**MapReduce:** is a programming model for processing and generating large datasets in a parallel and distributed manner on a cluster.