

(Unterschrift des Betreuers)



Diplomarbeit

A Web-Based System for Comparative Analysis of OpenStreetMap Data by the Use of CouchDB

Ausgeführt am Institut für
Geoinformation und Kartographie der Technischen Universität Wien

unter der Anleitung von
Univ.-Prof. Mag.rer.nat. Dr.rer.nat Georg Gartner und Univ.-Ass. Dipl.-Ing. (FH) Manuela Schmidt

durch
Markus Mayr
Gschaidstraße 10
A-4052 Ansfelden

Wien, 19.10.2012

(Markus Mayr)

Abstract

This thesis evaluates the possibility to construct a web application, which operates solely within a geospatially enabled CouchDB database. Its purpose is the matching of POIs from OpenStreetMap with a random set of points. To do so, a simplified matching algorithm is constructed and evaluated. Furthermore, a way to support legal decisions when using OpenStreetMap data is implemented.

CONTENTS

1	Introduction	1
1.1	Motivation	1
1.2	Targets of Research	2
1.3	Base of Research	4
1.4	Structure of the Thesis	4
2	Required Systems and Methods	5
2.1	Open Street Map	5
2.2	CouchDB	11
2.3	GeoCouch	17
2.4	JavaScript	17
2.5	General Structure of the Analysis Program	20
3	Legal Issues	23
3.1	Open Street Map	23
3.2	Comparative Datasets	27
3.3	Software Licenses	27
3.4	Impact on the Analysis Program	27
3.5	Result of this Chapter	29
4	Simple POI Matching	31
4.1	Situation	31
4.2	Known Procedures	32
4.3	Acceptable Results	33
4.4	Developed Procedure	37
4.5	Further Optimisation	43
4.6	Evaluating the Method	45
4.7	Result of this Chapter	48
5	Integrating the System	51
5.1	Circumstances	51
5.2	Implementing the System	52
5.3	Operation of the Web Application	53
5.4	Testing the System	56
5.5	Result of this Chapter	59
6	Conclusion	61
6.1	Results	61
6.2	Optimisation and Further Research	62

7	Appendix	63
7.1	Legal Texts	63
7.2	Source Code	79
7.3	Comparison Maps	88
7.4	Pictures of the Web Application	105
A	Bibliography	107

Acknowledgements

I want to thank...

- ... **my parents** for allowing me to follow my interests of study and providing support for its duration.
- ... **Katharina, Rainer** and **Melinda** for their mental support and proof reading.
- ... **Clemens Beer** from the Austrian company Tupalo for providing datasets.
- ... the **OpenStreetMap community** for interesting discussions about topics related and not related to this thesis.
- ... the **CouchDB / GeoCouch community** for providing fast feedback when needed.
- ... the **lecturers of my program of studies** for making my time at the Vienna University of Technology an interesting and varied experience.
- ... **Mag.rer.nat. Dr.rer.nat. Georg Gartner** and **Dipl.-Ing. (FH) Manuela Schmidt** for their support with this thesis.

INTRODUCTION

Content of this Chapter

This chapter contains an overview of the motives for choosing this topic, an introduction to the matters of research and hypotheses, and an outline of the topics discussed in this thesis.

1.1 Motivation

1.1.1 Personal Reasons

During my studies at the Vienna University of Technology I was confronted with different topics of different origin. These ranged from applied cartographic issues to theoretical and sometimes experimental systems. The intersection of various fields of study was always a particular point of interest for me.

I learned not only about relational databases but also about document based databases. The trend to web-mapping based systems was evident and there is always need for easily accessible tools and methods for analysis. To build a system like the one proposed by this thesis requires a combination of various different aspects which (among others) include user-interface design, legal concerns, open-source mechanisms, acquisition of data, performance tuning and documentation writing. [Haverbeke \[2011\]](#) says in his introduction that *“Programmers who refuse to keep exploring will surely stagnate, forget their joy, and lose the will to program.”* In my opinion, this is also true for research in general.

According to my personal research, CouchDB has not been used for a project of this kind yet, or at least has not yet been documented. The reason for the application of a CouchDB in this thesis differs from the usually called arguments like “scalability”, “performance” and “effort of programming” as they are stated by e.g. [Leavitt \[2010\]](#). This system provides other great benefits for web-based programming as is discussed in the following chapters.

Another factor that is of great importance to me is sustainability, which is reflected in this thesis as well. The availability of OpenStreetMap data does not underlie any commercial restriction, as can be seen in chapter 20 in [Ramm et al. \[2010\]](#). Therefore, as long as there are people who are willing to support the project, it will prevail. The realisation of this project as a web-application ensures ongoing interoperability between different computer-systems. The focus on storing the project as an entity in a database guarantees a means of distribution and availability.

1.1.2 Professional Reasons

Open Street Map in Research

A paper by Schmidt and Neis [2011] shows many different fields of research which are possible concerning OpenStreetMap. There is also a great need for “openness” in these areas. As proposed by their article, this thesis has been announced on the Research page of the OSM-Wiki [OpenStreetMap Wiki, 2012a] and a connection to the OSM community has been made. It is a primary goal of this work to build a system that is readily and publicly available. Finally, it is not only the result of an analysis that has been published, but also the means to elaborate on said analysis by the user. The resulting source-code of this thesis has been published as open-source at <https://github.com/scubbx/simplematcher> or can be found in the folder `source` on the SD card in the back of this thesis.

Haklay [2011] defines three main areas of research concerning OpenStreetMap:

With OSM: New topics of research may be found by working together with the OSM community.

About OSM: OSM is observed and its mechanisms and potential are evaluated.

Using OSM: OSM data is used to produce new knowledge.

A system like the one proposed by this thesis falls into the latter two categories (*about OSM* and *using OSM*), since OSM data is being analysed externally, but is also used to generate meta-information about its users.

The author of this thesis sees the coming need for systems that allow easy analysis of OSM-data, at the very least because of further research topics described by Haklay [2011] (e.g. “*Internal culture and impact on gender, exclusion and long term engagement, Spatial and temporal patterns of engagement and contribution, role of ‘silent’ mappers, Cultural differences and the impact on mapping*”). Also, existing analyses mainly concern the street network. Information about the quality or completeness of single points-of-interest is rare (see section *Known Procedures* (page 32)).

Web-Application Programming

Murugesan et al. [2011] say that “*We’re entering a new era of the Web [..]*”, while talking about smart-phones, RFID and various other topics but also referring to the Web 3.0. This implies the development and research of new programming styles and possibilities to solve certain tasks. Murugesan et al. [2011] simply state that “*New kinds of Web apps are on the horizon.*”. They also name some challenges that need to be taken care of, like technical development, operation, organization and social matters.

Using CouchDB

For the creation of a system of this kind, a relatively new technology is used. In the early stages of productive use of a new technique, it is especially important to evaluate the practical usability for specific scenarios. The database concept used in this thesis offers some advantageous functions and methods of usage, which make it interesting to work with. Of course, there are some flaws to be expected as well. This thesis will also test the suitability of this database for the system proposed by this thesis.

1.2 Targets of Research

First, the hypotheses are defined. The criteria by which each of them is evaluated are stated. In chapter 6 these are compared to the actual results of this thesis.

1.2.1 Hypotheses

Besides the main-hypothesis (formatted as bold), additional hypotheses have been defined to thoroughly and systematically cover all relevant aspects. These handle different topics. When compiling a multifaceted system like the one proposed, many different factors must be accounted for. The hypotheses are:

- **It is possible to construct a system to perform a simplified comparative analysis of specific OpenStreetMap – POIs with previously selected datasets by using a document-based database.**
- By paying attention to legal guidelines it is possible to declare the legal properties of the results programmatically.
- By using a spatially enabled document-based database and a well defined input dataset, a simplified geographical comparison algorithm can produce acceptable results.
- The complete system can be contained within one independent, single database and perform a large scale analysis in an efficient manner.

1.2.2 Criteria and Methods

To verify or falsify the hypotheses the following criteria and methods are used:

Legal Issues (Chapter 3)

Criteria: The system does not have to put restrictions on its output, but when necessary, these are well defined and visible to the user.

Method: Work through the different licenses, construct the system accordingly and make sure, information about restrictions reaches the user.

Simplified Matching (Chapter 4)

Criteria: The algorithm is useful and delivers acceptable results.

Method: Compare the result with manual checks of smaller areas by using a set of criteria and define “acceptable”. Evaluate the comparison algorithm.

System in a Database (Chapter 5)

Criteria: The system as well as the OSM dataset has to be stored within one single database. The system should be capable of performing up to national extent.

Method: Create and replicate the final system to another database and perform a check. Run the system on great extents and evaluate the performance of the preparation of data as well as the execution of the comparison.

1.3 Base of Research

Two different kinds of information were needed to compile this thesis: theoretical information and practical information.

Sources for the theoretical part were found on the “research” page of the OpenStreetMap wiki [OpenStreetMap Wiki, 2012] but also by general searches for scientific papers. For this the “Google Scholar” search engine [Google Incorporated, 2012] provided a solid instrument. Also, reference books and libraries proved to be a reliable source of information.

Practical information was acquired by remembering techniques learned during my studies, getting in touch with people responsible for specific projects and making good use of the aggregating and information collecting capabilities of the internet. Here, direct contact with people from the communities of the involved open-source projects proved to be very helpful.

1.4 Structure of the Thesis

The definition of the three sub-hypotheses also gives a general structure to this thesis. Each of these hypotheses is dealt with in a separate chapter. Additionally, a chapter to describe the techniques in use and a chapter for review are included (see *figure 1.1*).

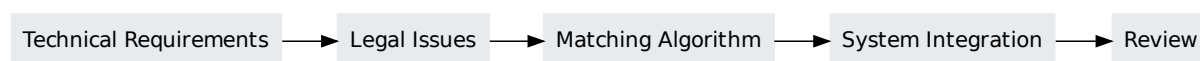


Figure 1.1: Topics

The system developed by this thesis can be tested by using the VirtualBox image that is saved on the SD card located at the end of the thesis.

REQUIRED SYSTEMS AND METHODS

Content of this Chapter

There are multiple techniques and systems involved in creating the program proposed by this thesis. Their prerequisites are the focus of this chapter.

While there are already other established systems used nowadays which could provide some of the selected techniques, it is important to try out new ways and possibilities to find out if these offer previously overlooked benefits. The selection of systems and methods used for this thesis is an elaborate one and goes in accordance with the targets of research.

The following sections give reasons for the choices made and explain necessary fundamentals.

2.1 Open Street Map

An overview of the history of the OpenStreetMap project is given and different methods of gaining access to its data are described. Also, the internal data structure is explained, since it is necessary to know about it to do analyses.

2.1.1 Origins and Meaning of the Project

As the history page of the OpenStreetMap-Wiki [OpenStreetMap Wiki, 2012b] says, the project was founded in 2004 by Steve Coast. It describes itself as “[..] *an editable map of the whole world (built largely from scratch) which is released with an open content license.*” [OpenStreetMap Wiki, 2012c]. It is a prime example of volunteered geographic information and has become a topic for numerous analysis about data quality.

Helbich et al. [2012] state that “[..] *future research is needed to get confidence, especially other reference datasets and more case studies must be analysed and other methodological approaches must be tested.*” . In accordance with the opinion of the author, this thesis represents such an alternate approach.

As can be seen in *figure 2.1* (page 6), the project is gaining publicity and is well maintained. It is not to be expected that this situation will change in foreseeable future.

It is therefore safe to assume that research concerning OpenStreetMap is of future importance.

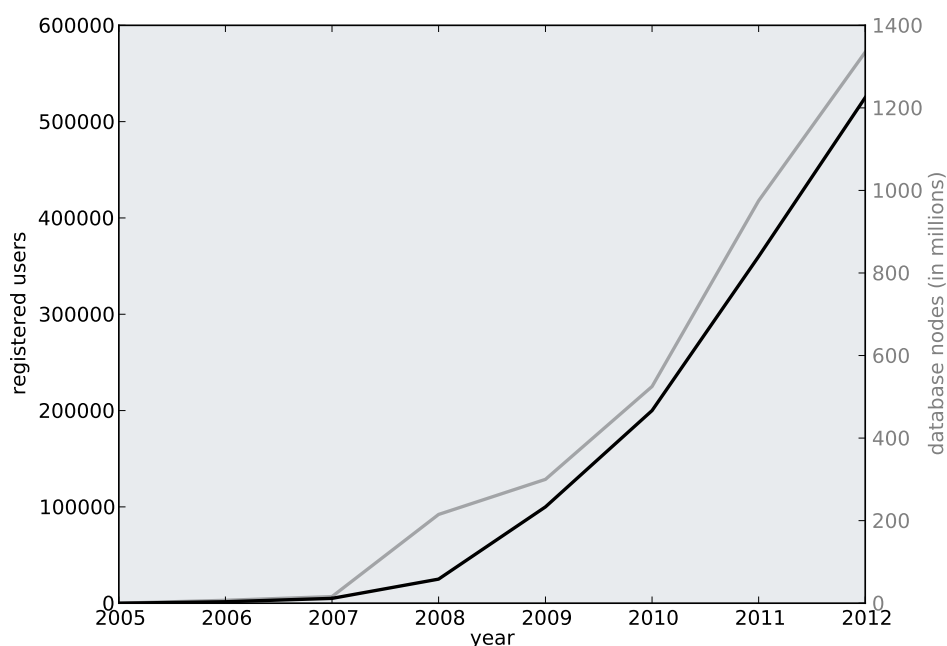


Figure 2.1: Development of OSM users and database content (generalised, source: see [OpenStreetMap Wiki \[2012d\]](#))

2.1.2 License

The license under which OpenStreetMap data is published is the Open Data Commons Open Database License (ODbL) [[Open Data Commons, 2012a](#)]. It states that whatever manipulation is made to the original OSM database, the manipulated version also has to be published under the ODbL. The resulting product may be treated in any way convenient to the creator (even as commercial product) as long as it does not resemble a database and the source database of this product is published, too. This topic is covered in more detail in section [Legal Issues](#) (page 23).

2.1.3 Data Model

Ramm et al. [2010] explain in chapter 6 that the data model allows a high degree of abstraction. The user has a lot of freedom in describing objects. Since OpenStreetMap does not limit itself to any field of interest, this is helpful but also challenging for applying general algorithms that require standardised input data. Luckily, as described in chapter 7 by Ramm et al. [2010], the community behind the OpenStreetMap project usually sticks to a set of self-imposed restrictions and rules concerning the description of data. New ways of describing objects are usually discussed before they are recommended to new mappers, as can be seen on the “Proposed Features” page [[OpenStreetMap Wiki, 2011](#)].

Geometry Information

Geographical data is saved as one of the following types: *node*, *way* or *relation*.

Nodes are the most basic type of entry and only consist of two geographic coordinates. They can only describe a single point by themselves.

Ways in contrast are an ordered list of nodes that encode the start, ending and vertices along a line. These nodes are stored as single nodes in the OpenStreetMap database. A way only consists of the `ids` of the nodes the way is composed of.

Relations are similar to ways. They are an ordered list of any entity contained in the OpenStreetMap database. These entities may even be relations by themselves. Each entry in the relation is assigned a `role` which describes the elements function. The relation itself may also be described by additional attributive information.

Typical situations for relations are bus routes (where a single route consists of multiple ways) or buildings with a court. While there is no definition of `role` necessary with a bus route (see [figure 2.2](#)), it has to be defined in the latter example (see [figure 2.3](#)): The line that describes the outer extent of the building is assigned a `role` value of `outer`, the line that describes the border between the building and the court is assigned a `role` value of `inner`. These roles may be mixed depending on the given situation (this may be the case when e.g. mapping an isle within a lake that is completely surrounded by forest).

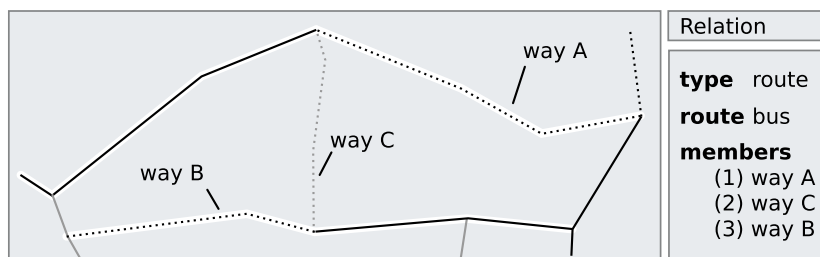


Figure 2.2: Example of a route relation
(dotted line) bus route, (black / grey lines) streets

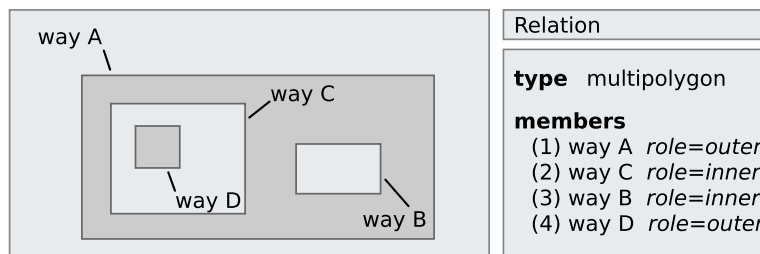


Figure 2.3: Example of a multipolygon relation
(dark grey) building

Attributive Information

Each entry in the OSM database can be tagged by an arbitrary number of key-value tags. For most of these tags there exists consensus in the OSM community on how and when to use specific keys and values. A bakery, for example, is indicated by the key `amenity`. The value for this key is `shop`. This is usually represented by `amenity = shop`. Since we do not only wish to say that our bakery is a shop but also want to state that it sells pastries, we have to add a further tag, namely `shop = bakery`. Let us say that this bakery also has a name which goes like "Mad for Bread". We will also include this information by using the tag `name = Mad for Bread`.

When we take a look at the complete definition for our bakery now, we will see the the attributes as listed in *figure 2.4* :

key	value
name	Mad for Bread
amenity	shop
shop	bakery

Figure 2.4: Attributive information defining a bakery

An overview over the most commonly used tags is given on a web-page called “TagInfo” [Topf, 2012a]. Additionally to these manually created tags there is some information that is always saved alongside the previously named attributes. This is an excerpt of an original entry of the OSM database which describes a gas-station:

```
<node id="223694456"
  version="7"
  timestamp="2011-07-17T11:41:10Z"
  uid="53306"
  user="Wolfgang B"
  changeset="8747023"
  lat="48.5641759"
  lon="16.5760244">
  <tag k="addr:postcode" v="2130"/>
  <tag k="fuel:cng" v="yes"/>
  <tag k="addr:country" v="AT"/>
  <tag k="name" v="AGIP"/>
  <tag k="amenity" v="fuel"/>
  <tag k="addr:street" v="Mitschastraße"/>
  <tag k="addr:city" v="Mistelbach"/>
  <tag k="operator" v="Agip"/>
  <tag k="addr:housenumber" v="38"/>
</node>
```

The information stored automatically is:

- `id`: holds a number unique to this element. It is possible to identify this element by only using this number.
- `version`: stands for the version of the object (when changing any information of an element, the old information is not lost, but a new version with the changed information of the same element is created). The element with the highest version number is the most recent. This number also gives information about how often an element has already been modified.
- `timestamp`: gives information about date and time when the modification of the element with this version took place.
- `user`: contains the user-name of the user which did the editing on this version of the object.
- `uid`: is the user-id of the previously explained `user`.
- `changeset`: is the id of the changeset the element is assigned to. A changeset summarises a group of changes that are done by one user at one time.

2.1.4 Access to OpenStreetMap Data

The data gathered by the OpenStreetMap project would not be of much use if it would not be easily accessible. This can be done by directly downloading the complete database or by requesting objects individually.

planet.osm

At the web-page <http://planet.osm.org> one can download the complete OSM coverage of the world in a file called “planet.osm“. Additionally, there are also offered so-called “*changesets*” that contain only the objects that were subject to a change since a specified date. By using these it is more convenient to update a local OpenStreetMap database.

These files are usually offered in a standardised XML format. Its specifications can be found at [OpenStreetMap Wiki \[2012e\]](#) . Another format is PBF which is a more recent binary file format that allows faster processing for external programs and is smaller in size but is not human-readable any more. The format specifications can be found at [OpenStreetMap Wiki \[2012f\]](#).

Full History Dumps

One problem with the previously described `planet.osm` file is that it only contains the most recent version of any OSM entity. When one is interested in information relating to older versions of an entry, this is not enough. Since April 2011 there are experimental dumps at the web-page <http://planet.osm.org/cc-by-sa/full-experimental/> available. They contain every version of every entity ever submitted to the OSM project. The downside to this approach is that it consumes a lot more space, since, depending on its editing-history, there may be more than e.g. 20 versions of one single entity. Another problem is that most software written to process OpenStreetMap data can not yet cope with history information. There exists an experimental tool called “osm-history-splitter” that can cut specific regions of space from the complete history-dump [[MaZderMind, 2012](#)].

OSM APIs

The most direct way to gain access to OSM data is to use OSM APIs (*Application Programming Interface*). They are implemented as REST interfaces and can be accessed by a wide variety of programming languages without the need to integrate additional methods. Besides the default API there are others that allow access to databases that are cloned from the original one on a regular basis. An overview over different APIs and their possible usage is given in [figure 2.5](#) (page 10).

Default API

This is the default API when requesting or editing OpenStreetMap data. It is well documented and directly linked to the main database of the OSM project. It is possible to request any kind of information available by using requests formatted like documented on its web page [[OpenStreetMap Wiki, 2012g](#)]. Detailed information about the default API is given in chapter 22 in [Ramm et al. \[2010\]](#). It is possible to request specific nodes, ways, relations or the history of any of these entities and changesets.

An example request for getting information about just one single node looks like this:

```
http://api.openstreetmap.org/api/0.6/node/1365240890
```

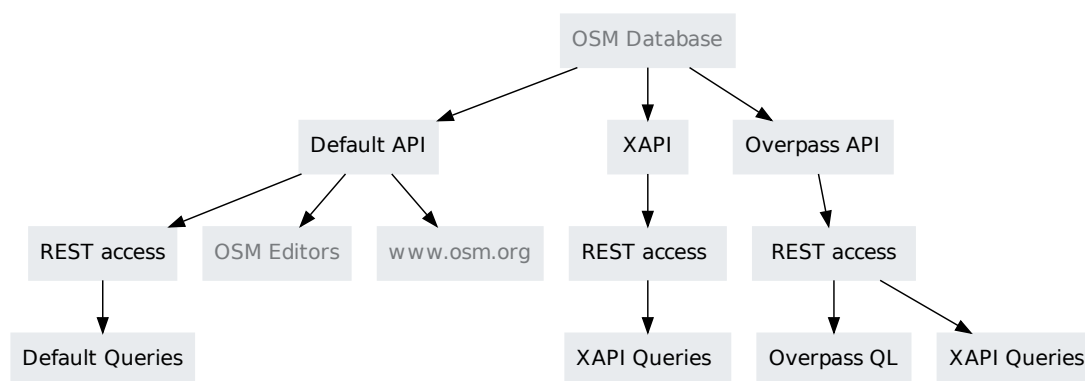


Figure 2.5: Overview over different OSM APIs
(source: see Olbricht [2012])

The results are presented in XML:

```
<osm version="0.6" generator="OpenStreetMap server">
  <node id="1365240890" lat="48.2068387"
    lon="14.2862104" version="1" changeset="8756205"
    user="ScubbX" uid="223638" visible="true"
    timestamp="2011-07-18T09:33:21Z"/>
</osm>
```

These than can be processed by typical XML parsers.

XAPI / jXAPI

The “OSM Extended API”, as also described by Ramm et al. [2010] in chapter 23.1, is an extended version of the default API. The main difference is that this one does only allow read-access to its database. The XAPI database is replicated with only a minutely difference from the main OSM database, so it is sufficiently up to date in most cases. The XAPI allows more advanced queries while the query-syntax stays mostly the same as with the default API. The most interesting feature of the extended API is its capability to deliver specific versions of not only nodes but also of ways.

A sample request using the XAPI service requesting all nodes within a given area looks like this:

```
http://jxapi.osm.rambler.ru/xapi/api/0.6/node[bbbox=
11.5411953,48.144,11.5411955,48.1445]
```

The resulting XML corresponds to the one generated by the Default API.

According to OpenStreetMap Wiki [2012h], the original XAPI servers are no longer active since January 2011. They have been replaced by a Java-implementation called jXAPI. The request-syntax stays the same, only the back-end has changed.

Overpass API

Olbricht [2012] describes the Overpass API in detail: The Overpass API again is a replicated version of the original OSM database. The difference is that anyone may install a new instance of the Overpass

API on a new server, so this greatly lifts the load on the Default API and the XAPI. It is designed to store data in a way optimised for queries and uses concepts that have been specially developed for its purpose (like an own database back-end). These are the reasons why it is very fast and still offers ways to do complicated queries. The Overpass API offers the possibility to use the XAPI query-syntax but also a query method on its own called “Overpass QL”. The latter one is, despite its more complex nature, still usable as a REST compatible interface.

Using the XAPI Compatibility Layer, a query looks like one made to the XAPI:

```
http://overpass.osm.rambler.ru/cgi/xapi?node[ bbox=
11.54185,48.1435,11.5419,48.1445]
```

The resulting XML contains less meta information about the requested entities (like `timestamp`, `user` or `changeset`) than results of the other APIs by default:

```
<osm version="0.6" generator="Overpass API">
  <note>
    The data included in this document is from
    www.openstreetmap.org. It has there been
    collected by a large group of contributors.
    For individual attribution of each item please
    refer to http://www.openstreetmap.org/api/0.6/
    [node|way|relation]/#id/history
  </note>
  <meta osm_base="2012-05-23T23:06:02Z"/>
  <node id="268098206" lat="48.1437736" lon="11.5418928"/>
</osm>
```

It is still possible to get these missing tags by directly requesting them.

The Overpass API is still under development but has, according to its author, already reached a stage of development where it can be used.

2.2 CouchDB

If not stated otherwise this section is mainly based on [Wenk and Klampäcker \[2011\]](#).

The CouchDB project is a rather young database management system. It was started in 2005 and became an Apache Software Foundation project [[Foundation, 2012](#)] in 2008.

2.2.1 Differences to RDBMS

CouchDB is a non-relational document-based database system. It belongs to the database type of “NoSQL” databases. This term stands for “Not Only SQL” or “Not Relational” [[Cattell, 2011](#)]. There are important differences to classical relational database-management systems (RDBMS). While there are various different NoSQL databases, this section will concentrate on the characteristics of CouchDB.

Data Storage

In a typical RDBMS, information is stored as tuples in a table (one tuple corresponds to one row). If the data becomes very complex, individual pieces may be split across multiple tables [[Miler et al., 2011](#)]. When making special queries, selected tables are joined again to provide the wanted information. When

storing data like that, one is restricted by its schema (see *figure 2.6*). In most cases it is not advised or at least not practical to just add or remove a column.

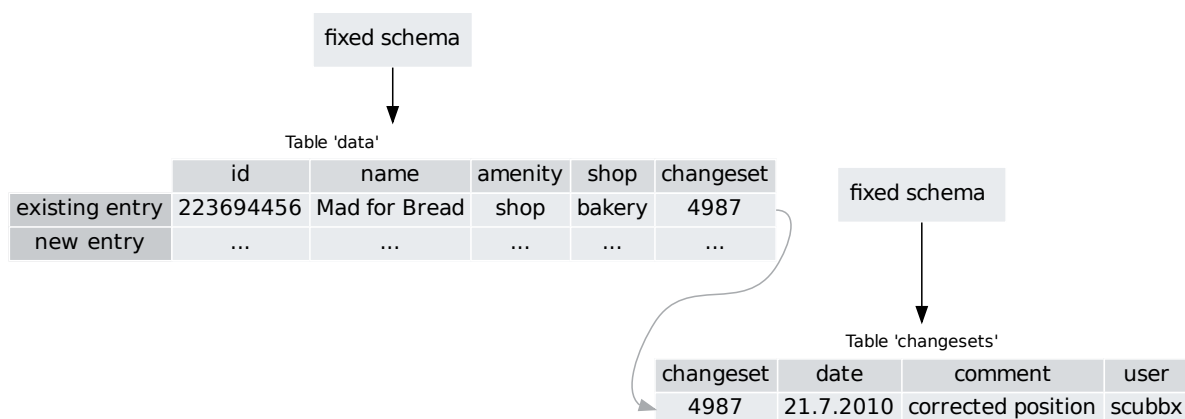


Figure 2.6: RDBMS

This system is well proven and there exist examples to almost every case one can think of. While this is a good argument, there are special situations where some of the benefits become a hindrance. This is, for example the case when one large table is split up to different machines on the same network and has to be joined again for a specific request. Then all the machines holding a share of the table have to be reached for the JOIN-operation which slows down the query.

Leavitt [2010] say “*Relational databases work best with structured data [...] which readily fits in well-organized tables. This is not the case with unstructured data [...] .*”. In a document-based database, one entry, which loosely corresponds to a tuple, is a “document” (*figure 2.7* (page 12)). In CouchDB this document is populated by an arbitrary number of *key-value* pairs. These pairs contain all the data and are not specified prior to the creation of each single document. This means that there exists no restriction on the schema of a document on the database-side. Any time one can add an additional key or delete it.

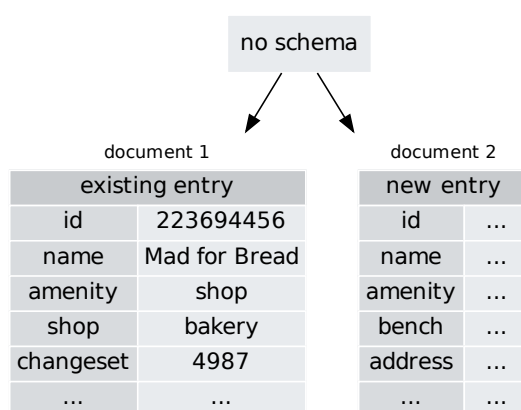


Figure 2.7: Document based database storage

In a classic RDBMS it would be necessary to add an additional column to the table to store the wanted value, but then all the other entries also have to assign a value to this column (even if it is just a *NULL*-value).

Distribution

One of the main reasons NoSQL databases are chosen in production are their capabilities concerning distribution and replication of data ([Leavitt, 2010] and [Miler et al., 2011]). While it may be of importance when further development of the analysis program is done or it is connected to other databases, this is not of primary concern to this thesis.

Nonetheless, it should at least be said that, as stated by Miler et al. [2011], CouchDB offers a very convincing way to replicate its database automatically to stay synchronized with other versions of its database. This is especially useful with mobile applications, when there is risk involved to lose the connection to the database.

According to Salo [2012], there is also ongoing development in designing a version of CouchDB that scales its databases to multiple nodes called “BigCouch”. Since document-based databases do not require JOIN operations by nature, these kind of requests will not slow down a query when executed on a distributed database.

System Requirements

The system requirements for databases are comparatively low for CouchDB since it performs the most demanding operations directly on data stored on the hard drive. Because of this, hard drive space is the one thing that is consumed most.

CAP-Theorem and Eventual Consistency

[Gilbert and Lynch, 2002] formulate the CAP theorem which describes three different properties of distributed systems (see *figure 2.8* (page 13)). Only two of these can be fulfilled at once.

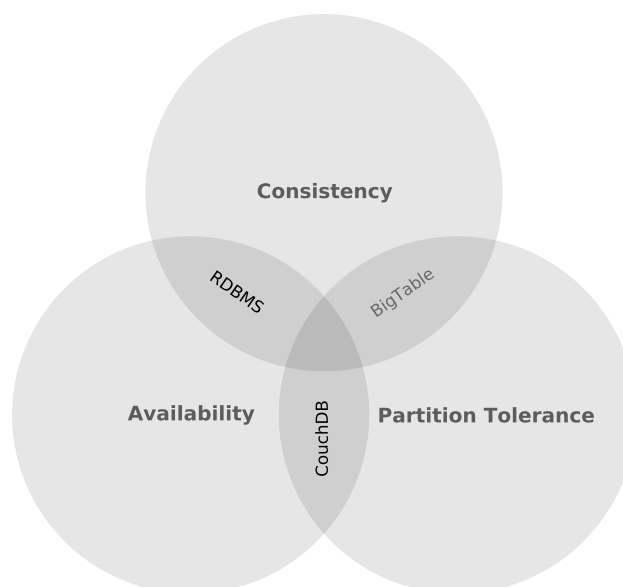


Figure 2.8: The CAP theorem
(source: see Anderson et al. [2010])

“C” stands for Consistency: The complete database has to reflect a single state. All reads from different partitions of the same database must return the same values. The same goes for write operations.

“A” stands for Availability: The system has to be able to respond.

“P” stands for Partition Tolerance: This means, that when a node of a distributed database is lost, the whole system has not to be affected by this.

CouchDB fulfils “Partition Tolerance” and “Availability”. “Consistency” cannot be achieved completely. Since it is not possible to guarantee that all different nodes contain the same information at once without a short blackout of the node that has to be updated, this would result in a violation of “Availability”.

To compensate for this, CouchDB guarantees “Eventual Consistency” (Orend [2010] in chapter 2.3). This means that it is only guaranteed that all partitions of a database will eventually contain the same data. As Orend [2010] says in chapter 2.2, RDBMS use a central lock-manager to assure ACI (Atomicity, Consistency, Isolation), at least partially. In a distributed database, this locking poses great problems, since all servers have to contact a central lock manager each time an operation is performed. To oppose this, CouchDB uses a multiversion concurrency control mechanism (MVCC) instead of locking mechanisms. This means that version information saved alongside with each entry is evaluated and only if it matches certain criteria (like a high or low enough version number) the operation is allowed. This mechanism is described in greater detail in e.g. Orend [2010] in chapter 2.5 .

In the prospect of this thesis this plays a minor role, since it is not dealing with distributed databases, but when expanding the system proposed by this thesis, it could become relevant.

2.2.2 Data Storage - JSON

CouchDB uses the JSON (JavaScript Object Notation) format for storing, inputting and outputting data. This format is specified in [Crockford, 2006] and actually is a subset of the JavaScript language. This is of great use to programmers when they prefer JavaScript to program the software that is interacting with CouchDB since it is directly compatible. A typical CouchDB-document looks like this:

```
{
  "_id": "node1066619375",
  "_rev": "1-d84091df8d07795ee1a00b8e12068b31",
  "geometry": {
    "type": "Point",
    "coordinates": [
      16.3506774,
      48.2094443
    ]
  },
  "type": "Feature",
  "properties": {
    "changeset": 6772558,
    "uid": 56986,
    "tags": {
      "amenity": "bank",
      "atm": "yes",
      "name": "Oberbank"
    },
    "timestamp": "2010-12-26T21:39:00Z",
    "version": 1,
    "user": "alice23"
  }
}
```

As can be seen, it may also contain *lists* and *arrays* besides simple JavaScript data-types. Two data fields always have to be present in a CouchDB document. One is `_id` which stores a unique identifier of the document and the other one is named `_rev` which holds the revision-number. The number of revision is especially important for the MVCC mechanism and is modified each time the document is changed.

A simple change to a document always results in an update of the complete document. It is possible to change just one letter, but the whole document is updated and a new number of revision is assigned to it.

The single documents themselves are stored on disk organized in a B+-Tree.

Attachments

There is one feature that deserves special attention. It is possible to store files of whatever type as attachments to documents in CouchDB. This allows for inclusion of any data possible into the database. These attached files are reachable by a unique URI. This function is of great importance when storing complete web-applications inside CouchDB.

2.2.3 Communication with CouchDB - REST

To input and output data, one has to make requests. A powerful database system also has to allow to specify some kind of query.

Making Requests

To interact with CouchDB one makes simple requests via a Representational State Transfer (REST) interface. [Bayer and Sohn \[2007\]](#) describe the WWW itself as a giant REST-application, which gives an idea about what is possible to achieve by using this method. REST describes a way how to access resources by using the HTTP-protocol.

A REST request consists of a unique URI (Uniform Resource Identifier) to which the response is always the same. On the WWW this corresponds to the URL which addresses a specific resource. Hereby the URI has to contain all the information necessary to execute the request. The most simple query to the CouchDB database-server (which is running on the local machine on port 5984 in this example) looks like this:

```
GET http://127.0.0.1:5984
```

This results in the response of some meta-information about the currently running CouchDB instance. When asking for data directly, the response is a JSON document containing a list of documents that have been requested. It is possible to get access to every individual document or its attachments by using its corresponding URI. For more detailed information about how to make specific requests, chapter 1.1.2 in [Wenk and Klampäcker \[2011\]](#) or chapter 4 in [Anderson et al. \[2010\]](#) are recommended.

Since it is possible to access CouchDB resources via an URL, CouchDB is not only a database, but also a web server.

Views - Map/Reduce

Instead of using SQL to make complex queries on a CouchDB, one has to program separate `map` and `reduce` functions (see [figure 2.9](#)). This is a completely different concept than including all directives

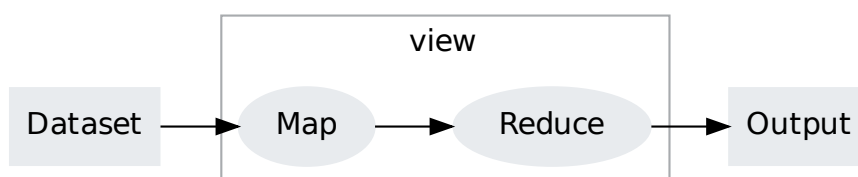


Figure 2.9: Complex queries in CouchDB

in the request (like it is done by using SQL). These two functions compose a *view* which can be called after being defined. It is not necessary to create both functions, often the `map` function suffices.

The map function is a function that filters and sorts all available documents by executing its code and applying it to each document in the database. The result is a list of sorted key-value pairs where the code of the function describes what will be the output for each processed document (see [figure 2.10](#)).

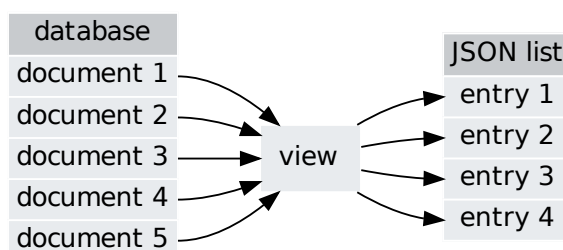


Figure 2.10: Role of a view

The function can be programmed in different languages. By default this language is JavaScript but there are possibilities to extend CouchDB with other interpreters. In its simplest form a map function looks like this:

```
function(doc) {  
    emit(null, doc);  
}
```

As said before, this function is executed on each document in the database. It does not evaluate anything, but only executes the `emit` command which resembles a `return` function and is used to produce an entry in the output JSON-list. The `emit` function needs two arguments, the first being the key, the second being the value. The variable `doc` represents the currently processed document as JSON. To allow a fast query, an index is generated and stored in a B+-tree. So, there exists an own index for every view that has to be updated when the view is changed. After a view has been defined, it is possible to call it by requesting it by its URI. Views themselves are also stored inside a CouchDB database document which then is called a “Design” document.

An optional reduce function takes the output of the map function and processes it in any way programmed by the user (usually it will do some kind of grouping, if used).

2.2.4 CouchApp

As we have described before, it is possible to store any file thinkable of as an attachment to a CouchDB document. These attachments then can be accessed by their URI. If we store a file called `index.html` as an attachment and open it with the URI `http://127.0.0.1:5984/_design/tempDesignDocument/tempApp/index.html` in a web-browser, it will open this very `index.html` and render its content. Following this procedure, it is possible to store every component needed for a web application inside a CouchDB, even its programming logic as `*.js` files and address it by its URI.

There exists a tool named “CouchApp” that facilitates the creation of HTML5 applications inside a CouchDB. It does this by taking a directory and transforming its content to a CouchDB design-document. As Anderson et al. [2010] say in chapter 10, by using CouchApps one does not “*need to worry about picking a language or framework; we’ll just work directly with couchDB [..]*”.

2.3 GeoCouch

To perform spatial queries, Mische et al. [2012] has developed a plugin for CouchDB called “GeoCouch” that mainly allows storing indices not only in B+-trees but uses R-trees. The implementation of this R-tree is called V-tree in this case as is described by Mische [2010]. The reason for using an R-tree is that this allows indexing of two dimensional geometry. If a so-called spatial index has been generated on a database, the user can do bounding-box requests in an efficient manner. This spatial index is defined by a “spatial function” (a spatial view) which describes where to look for the stored geographic coordinates. This function is also stored in a design document under the entry named “spatial”. A typical bounding-box request on such a spatial index looks like this:

```
GET http://127.0.0.1:5984/_design/tempDesignDocument/tempApp/  
_spatial?bbox=0,0,180,90
```

The GeoCouch plugin is currently under heavy development and new features are bound to appear. The version used is an experimental branch of GeoCouch that allows requests not only by bounding boxes but also polygons [Mische, 2012].

2.4 JavaScript

JavaScript is a scripting language originally designed in the year 1995 for laymen to extend HTML pages with effects [McFarland and Fröhlich, 2009]. Since then JavaScript has come a long way to its appearance as a powerful and widely used programming language.

JavaScript has to be used since the program, as described by this thesis, is a web application. The reason for this is that this already covers some of the desired targets of research by system design (e.g. user friendliness, presentation of the results, availability). This topic is discussed in greater detail in section *General Structure of the Analysis Program* (page 20). Also, according to Ousterhout [1998], “[..] scripting languages are designed for gluing: They assume the existence of a set of powerful components and are intended primarily for connecting components.”. Since this system will use independent components that will be “glued” together, this fits as well.

According to its creator TIOBE Software BV [2012], the TIOBE Programming Community Index is “[..] an indicator of the popularity of programming languages” . It is mainly composed of percentages of the amounts of web-search results of selected programming languages for the purpose of rating their popularity. *Figure 2.11* shows the TIOBE Index for a selection of well known programming languages.

It can be seen that in contrast to other widely used programming languages the index for JavaScript stays rather constant. This is no surprise since it is a reasonable choice when programming web applications which in turn are gaining importance.

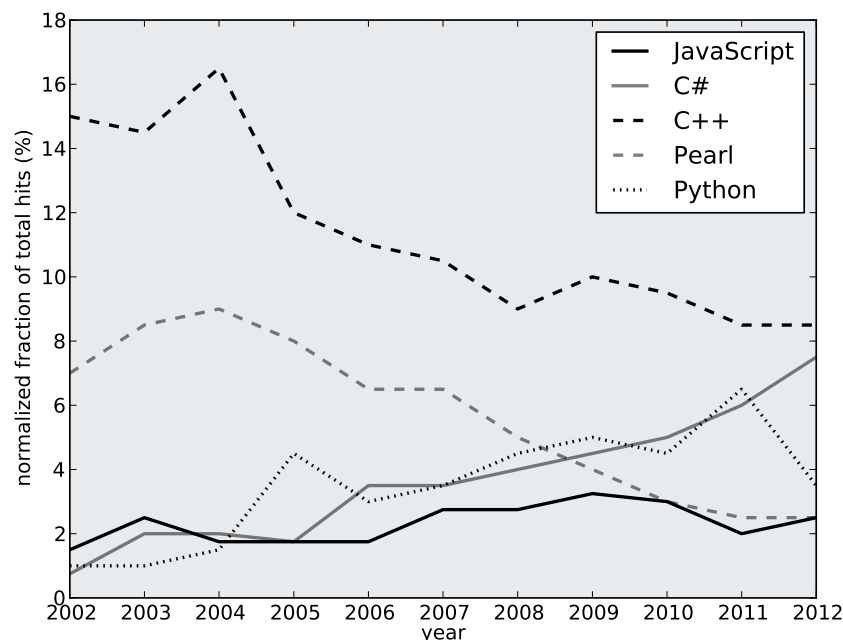


Figure 2.11: TIOBE Programming Community Index (generalised, source: see TIOBE Software BV [2012])

Delorey et al. [2007] state that the TIOBE Index is only an indirect method of measurement and propagate a more direct approach. They have compared data about 9997 open source projects which are hosted on SourceForge. This gives, amongst other things, information about the “[...] *relative popularity of the ten most popular programming languages over time.*”. So-called language profiles, which may mix two different languages, have been created since with some programs it is unavoidable to use more than one language. The popularity of some of these language profiles concerning different open source projects is shown in [figure 2.12](#) (page 19).

This shows a slightly different view. According to this calculations, at least the combination of JavaScript and PHP is used more often over time.

Other indications for the constant or even growing importance of JavaScript may be its recent use in offline scripting. The game engine “Unity 3D” [Unity Technologies, 2012] for example can be programmed by either C# or JavaScript. There also exist frameworks that allow application programming in JavaScript without the need to embed the code in HTML. One of these is the “Node.js” framework [Joyent Incorporated, 2012]. Haverbeke [2011] states in the introduction to his book that this framework “[...] *has recently been attracting a lot of attention.*”. CouchDB itself serves as another example for the execution of JavaScript code not interpreted by a web browser: Views may be programmed in JavaScript (which then are executed on the CouchDB server) as described in chapter 2.5.3 by Wenk and Klampäcker [2011].

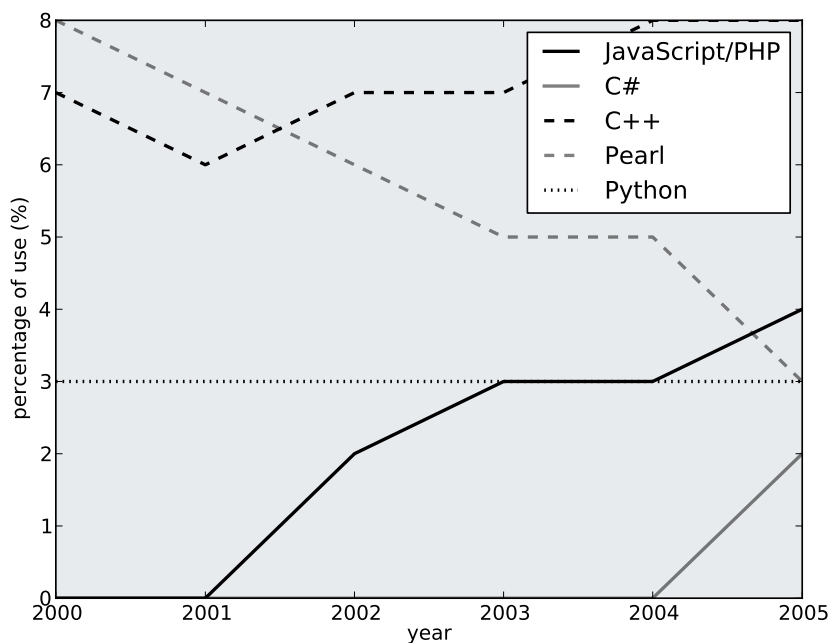


Figure 2.12: Language profiles by year and author (source: see Delorey et al. [2007])

2.4.1 Web Workers

“Web Worker” is a technique introduced with HTML5 and is specified by Ian Hickson, Google, Inc. [2012]. Figure 2.13 shows that support for web workers is well-implemented in recent browsers. Only Internet Explorer and the Android Browser do not support web workers at this time.

	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Opera Mobile	Android Browser
		9.0						10.0	2.1
	6.0	10.0				3.2		11.0	2.2
	7.0	11.0	17.0			4.0 / 4.1		11.1	2.3
	8.0	12.0	18.0	5.0		4.2 / 4.3		11.5	3.0
current	9.0	13.0	19.0	5.1	11.6	5.0	5.0 / 6.0	12.0	4.0
future	10.0	14.0	20.0	5.2	12.0				

Figure 2.13: Support for web workers in different web browsers (white): not supported, (grey): supported, source: see Deveria [2012]

A web worker is JavaScript code that is executed in parallel to the main JavaScript of the current web page. This lets the main page stay responsive and makes use of multi-threading.

To use it, one has to define a separate JavaScript file. The communication between the main-page and the web worker is happening by so-called messages. This message can be any string value and is the only way to move information from and to the web worker.

2.4.2 Libraries

Different libraries are used to facilitate development. These are:

- *OpenLayers* [OpenLayers Contributors, 2012] is used to display a map and perform projective calculations.
- *jQuery* [jQuery Foundation, 2012] is a library that, besides offering additional functions, facilitates development with JavaScript by offering a common interface to functions implemented differently by various web browsers. Moreover, CouchApp uses it.
- *Sparklines* [Watts, 2012] has programmed a JavaScript library which is used to generate diagrams from variables.

2.5 General Structure of the Analysis Program

As stated in *Hypotheses* (page 3), the analysis system proposed by this thesis uses a CouchDB database.

2.5.1 Web-Application

Since the complete system should be contained within one single database, a web based approach to implement the analysis program has been chosen. The reasons for this are:

- In general, CouchDB is very “web-friendly”. This means, that integral components of CouchDB are based on common web technologies and its API is also easily accessible from any web application (as has been shown and explained in section *CouchDB* (page 11)). There also exists a framework for developing web applications with CouchDB - namely *CouchApp*, as has been explained in section *CouchApp* (page 17).
- CouchDB is not only a database but also a web-server and web-proxy (Wenk and Klampäcker [2011], chapter 2.12).
- A system programmed as a web application is very system independent. Its only requirement is a recent web browser. Thus, it is easily-accessible from various computer-systems. Charland and Leroux [2011] describe a system for mobile application development that only depends on the web browsers of different mobile operating systems to more easily support a broad variety of mobile platforms. They say, that “[...] the only thing these operating systems have in common is that they all ship with a mobile browser that is accessible programmatically from the native code.”. This is the only interface they need. This is partially valid also for applications not developed specifically for mobile deployment. But, in the current case, this interoperability is only true for the part that interacts with the final user, since, as will be shown in section *Integrating the System* (page 51), there are other tools needed to prepare or extend the analysis system.
- Web applications are prone to produce an attractive user-interface design (or at least the design can easily be changed afterwards). By nature they allow a high degree of interaction and are still optimised for presentation.

2.5.2 Two-Tier Architecture

Miler et al. [2011] describes the method of building a web application by using a CouchApp as a “Two-Tier Architecture”. This stands in contrast to the classical “Three-Tier Architecture” as stated

by Ramirez [2000]. *Figure 2.14* (page 21) gives an overview about the difference of these two methods.

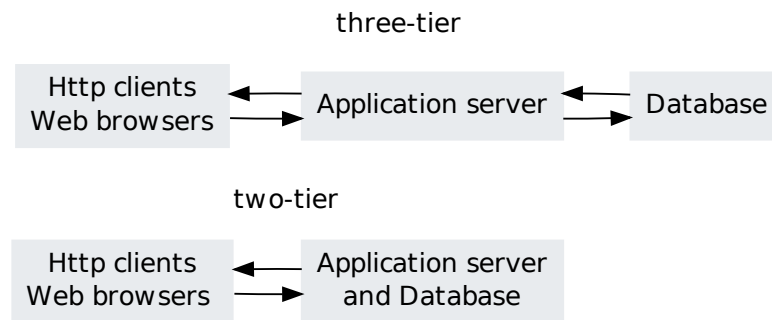


Figure 2.14: Three-tier vs. two-tier architecture
(source: see Miler et al. [2011])

A two-tier architecture reduces the number of techniques involved and therefore also reduces the requirements and error-proneness of the whole system. Actually, since CouchDB can also serve as a proxy-server, yet another function is accounted for.

2.5.3 Modular Programming

The analysis-program is structured in a modularized way. This means that the program code is separated into mostly independent modules according to their function, as Gumm and Sommer [2002] propose on page 189.

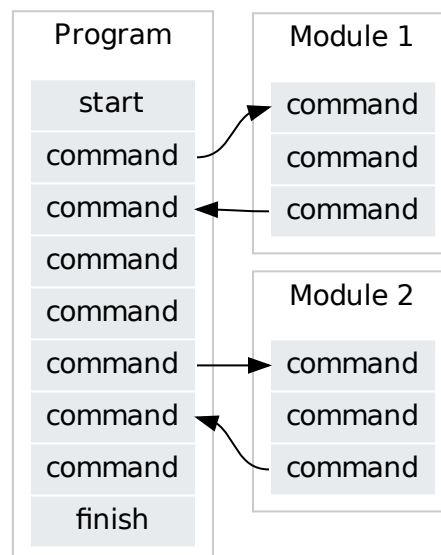


Figure 2.15: Modular programming architecture

Bain [2009] describes the benefits of modular programming. Some of these are:

- The code is easier to understand.
- Single modules may be reused in other applications.
- Program code can be separated in multiple files.
- The design of programs using a modular architecture is easier and errors are found faster.

In case of the analysis program at hand, a modular design additionally allows the separation of program code according to its topic as they are described by the various chapters of this work.

Some practical guidelines about how to achieve modularity with JavaScript are given by [Haverbeke \[2011\]](#) in chapter 7. He says that JavaScript has no built-in mechanisms to define modules. Therefore, it is up to the developer to structure his program accordingly.

Interface - GeoJSON

The single modules have to communicate with one another. Since CouchDB is using the JSON format to store information, it is only logical to stick to this standard. As [Butler et al. \[2008\]](#) claim in their abstract, “[...] *GeoJSON is a geospatial data interchange format based on JavaScript Object Notation (JSON). [...]*”. Therefore the GeoJSON format is used to store and exchange information between the modules when geographic content is processed.

LEGAL ISSUES

Content of this Chapter

Using data usually involves some kind of legal restrictions. In this chapter possible license issues are discussed and their importance for and impact on the analysis system is evaluated.

When mixing data from different sources, their licenses define the legal properties of the end product. The analysis program will use data from two different sources: OpenStreetMap and one other variable source. For testing the system, a commercial dataset is used alongside OSM data to perform an evaluation. To be aware of any legal implications when working with OSM data, the legal restrictions which apply to these datasets are discussed. Relevant licenses can be found in section *Legal Texts* (page 63) in the appendix.

Despite the legal boundaries defined by the used licenses, the Open Street Map community wants its data to be put to good use. In the experience of the author who has contact to the Open Street Map community, the licenses mainly exist to protect the project from commercial misuse and to create a legal foundation to argue upon.

This chapter represents an interpretation of the treated contracts.

3.1 Open Street Map

When the OpenStreetMap project was founded it was decided that it will use the *Creative Commons License* to ensure legal availability of gathered data. The problem with this license is that it exists for protecting creative work and not databases. This has led to certain difficulties (see section *Legal Problems* (page 24)). Therefore, the license was changed to the *Open Database License* in Summer 2012 which suits the legal needs of the OpenStreetMap project in a better way. To provide the Open Street Map Project with a stable background, the Open Street Map Foundation [Open Street Map Foundation, 2011c] was founded. The foundation is a non-profit organization to which membership is not restricted and whose purpose is to foster the Open Street Map project [Ramthun, 2012].

The following sections give an overview on the licenses in connection with the OSM-project. Additionally, *figure 3.1* displays the structure of the current license models.

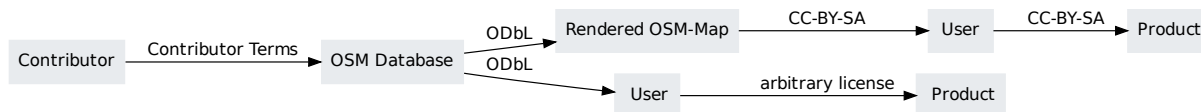


Figure 3.1: Actors, media and licenses involved in OpenStreetMap

In general, none of these licenses require the user to distribute his work based on OSM-data, but state the actions or license the user has to use if he decides to distribute it. A more detailed insight into legal issues concerned with OpenStreetMap Data is given by [Altenstein \[2011\]](#).

3.1.1 Contributor Terms

The *Contributor Terms* define the relationship between community members who add data and the OpenStreetMap itself. They are formulated by the [Open Street Map Foundation \[2011b\]](#). Every contributor has to accept these conditions which allow the OpenStreetMap Foundation to publish the entered data under an open license. Basically, these terms say that the user has to have the right to deliver his contributions to the Open Street Map Foundation, whereas the foundation has the right to publish the data and do whatever is necessary according to the current license.

In short, the Contributor Terms are about entering data into the Open Street Map Database.

3.1.2 Creative Commons (CC)

The license which was used until summer 2012 and was valid for the database itself as well as for the map tiles on the OSM-homepage, is the Creative Commons Attribution-ShareAlike 2.0 license (“CC-BY-SA 2.0”) [[Creative Commons, a](#)]. The [Open Street Map Foundation \[2011a\]](#) gives an overview of this previously used license and its conditions when using OpenStreetMap data. As is stated by [Creative Commons \[b\]](#), the CC-BY-SA license basically requires the user to:

“**[..] attribute the work in the manner specified by the author or licensor**” The [Open Street Map Foundation \[2011a\]](#) defines the way one has to attribute OpenStreetMap data by using the line “*Map data (c) OpenStreetMap contributors, CC-BY-SA*”. If one uses only the pre-rendered map tiles the attribution has to be “*(c) OpenStreetMap contributors, CC-BY-SA*”.

“**[..] distribute the resulting work only under the same or similar license to this one**” If distributed, the final work produced by the user has to use the same or a corresponding license. This implies that the user may actually sell his work, but he cannot prevent it from being given away at no cost by the purchaser.

These terms only apply to the complete end-product, when the user has generated a **derived** work. The counterpart to this is a **collective** work (see [figure 3.2](#) (page 25)). The definition of this phrases is discussed in greater detail in [OpenStreetMap Wiki \[2010\]](#). A derived work can only stand on its own and must be licensed as a whole. A collective work has a separate part of OSM-related information. As [Creative Commons \[a\]](#) say in section (4a), “*[..] this does not require the Collective Work apart from the Work itself to be made subject to the terms of this license*”. [OpenStreetMap Wiki \[2010\]](#) e.g. gives the example of a simple map included in a web-page which only constitutes a collective work, so only the map-part of the page has to comply with the CC-BY-SA. However, a t-shirt with an OpenStreetMap map printed on it has no separate OSM-part and can only be handled as a whole - this would be a derived work.

The pre-rendered tiles which are shown at the homepage of the OpenStreetMap project are available under the CC-BY-SA license.

Legal Problems

In chapter 20.6, [Ramm et al. \[2010\]](#) mention some problems the CC-BY-SA poses when being used in conjunction with OpenStreetMap data:

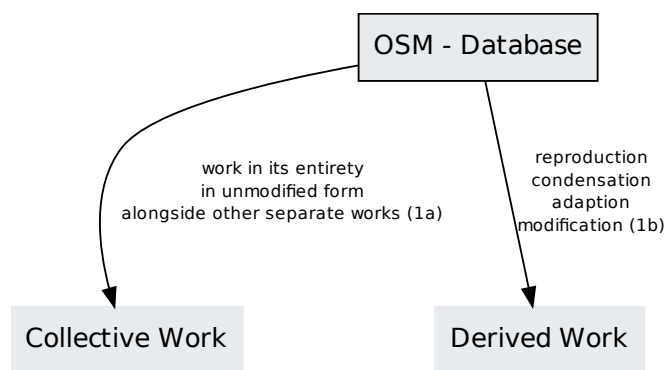


Figure 3.2: Legal structure of the CC-BY-SA
(the numbers refer to the sections inside the legal text)

- In some countries the legal conception of “facts are free” exists. This would result in the loss of the license of the OpenStreetMap data in this country.
- There are discussions whether or not copyright is applicable to geo-data at all. If not, then the CC-BY-SA would lose its meaning.
- The CC-BY-SA requires its application on the end-product but the OpenStreetMap project wants to secure its database. The intention for choosing a free license was that any edits made to the original dataset should also be entered in the OpenStreetMap database. Therefore, this license actually does not fit very well.

These are the main reasons why the OpenStreetMap database changed its license to the ODbL (see next section).

3.1.3 Open Database License (ODbL)

With the introduction of the Open Database License (ODbL) [Open Data Commons, 2012a] the use of OpenStreetMap data was facilitated considerably. The new license focuses on the database and not a product. *Figure 3.3* (page 26) shows a diagram of some rights and restrictions made by the ODbL (this figure represents an interpretation of the ODbL by the author). The user is allowed to give the product produced out of OSM data any license desired. As shown by the summary of the ODbL on [Open Data Commons \[2012c\]](#), the user now has to do the following:

“**[..] attribute any public use of the database, or works produced from the database [..]**” The attribution is required not only for any produced database but also for products (“work”) created from it.

“**[..] If you publicly use any adapted version of this database [..] you must also offer that adapted database under the ODbL.**” When applied to a database, this applies to the database that was used to produce the final product. Only this last database (or the methods to reproduce it) in a step of processes has to be published under the ODbL.

“**If you redistribute the database, [..] redistribute a version without such measures.**” (DRM) One is allowed to use DRM measures, as long as there also exists a freely available version of the same database.

Again, one has to make a distinction between *Collective Database* and *Derivative Database*.

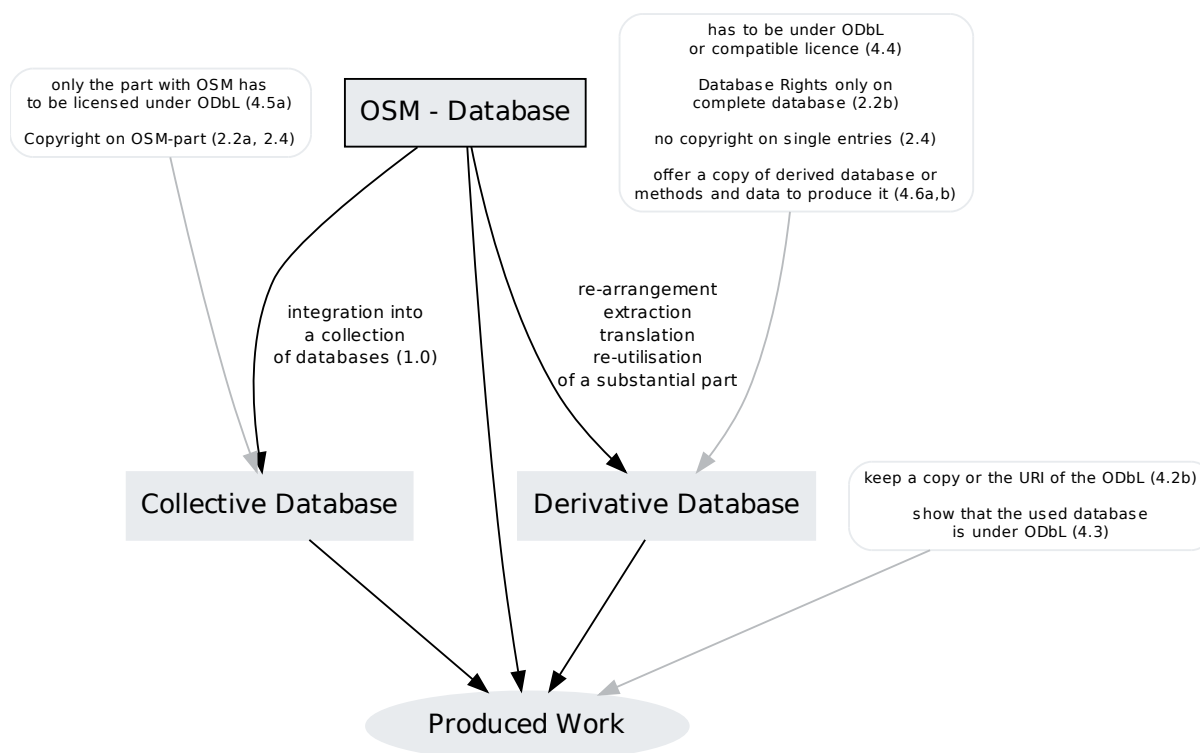


Figure 3.3: Legal structure of the ODbL
(the numbers refer to the sections inside the legal text)

Collective Database “Means this Database in unmodified form as part of a collection of independent databases in themselves that together are assembled into a collective whole.” (Open Data Commons [2012a], section 1.0)

Derivative Database “Means a database based upon the Database, and includes any translation, adaptation, arrangement, modification, or any other alteration of the Database or of a Substantial part of the Contents. This includes, but is not limited to, Extraction or Re-utilising the whole or a Substantial part of the Contents in a new Database.” (Open Data Commons [2012a], section 1.0)

Insubstantial Use

As Ramm et al. [2010] claim in chapter 20.7, an “insubstantial” use of contents of the OSM-database is not an issue with regard to the restrictions posed by the ODbL. As stated in 1.0 in the ODbL, substantial “Means substantial in terms of quantity or quality or a combination of both.” In OpenStreetMap Wiki [2012i] a description is given about what exactly is understood by the term “insubstantial”:

- “Less than 100 Features“
- “More than 100 features only if the extraction is non-systematic and clearly based on your own qualitative criteria [..]“
- “The features relating to an area of up to 1,000 inhabitants [..]“

Database content License (DbCL)

The ODbL specifies the legal properties of the Open Street Map Database in the context of a database (Open Data Commons [2012a], Preamble). The Database Content License (DbCL) [Open Data Com-

mons, 2012b] describes the usage of the content protected by the ODbL. According to Altenstein [2011] this is necessary if any database license is not applicable (e.g. due to legal constraints in specific countries). In that case the DbCL (among other legal details) basically defines that everyone using the database has to comply with the rules of the ODbL (“*You must comply with the ODbL.*” [Open Data Commons, 2012b]). Thus it is ensured that the content of the ODbL is valid under any circumstances.

3.2 Comparative Datasets

There will be cases when the comparative dataset will be made available only for the viewer itself (like a user with data from a company who does not wish these data being published under a free license). In case of this thesis, data from Tupalo [Tupalo Internetservices GmbH, 2012] is being used which is made available only for use in this thesis. It is not allowed to be distributed or publicly integrated into any other dataset.

3.3 Software Licenses

When specific software is saved alongside OSM data inside the same database, one might think that it is necessary to think about legal consequences since the database itself has to correspond to the rules specified in the ODbL. The solution is quite simple as there is a difference between the technical and the legal definition of a “database”. The ODbL defines a “Database” the same way as it is defined by the European Union [1996]: “*A collection of material (the Contents) arranged in a systematic or methodical way and individually accessible by electronic or other means [..]*”. This Database Directive is integrated into Austrian law by the “Bundesgesetzblatt Nr. 25 / 1008” [Bundeskanzleramt Österreich, 1998].

One can legally separate the program code from the actual data. The Bundeskanzleramt Österreich [1998] for example says in §40f “*Ein Computerprogramm, das für die Herstellung oder den Betrieb einer elektronisch zugänglichen Datenbank verwendet wird, ist nicht Bestandteil der Datenbank.*” (translation: *A software-program made for the creation or service of an electronically accessible database is not part of this database.*).

3.4 Impact on the Analysis Program

The analysis program itself needs pre-rendered map-tiles as well as data from the OpenStreetMap database (see figure 3.4 (page 28)). Additionally, data from another source is used to perform the comparative analysis. Therefore it is important to think about how to construct the analysis-program in a way that corresponds to the CC-BY-SA, the ODbL, as well as to the limitations that come with the comparative dataset.

3.4.1 Personal Use

If the results of the program are used only by the viewer in a personal context, none of the above restrictions concerning data from OpenStreetMap apply, since they are only valid if the data is used in a public context. There might only be restrictions based on the comparative dataset, but since the results are only viewed by the same person that holds the rights to use this dataset, there is no problem.

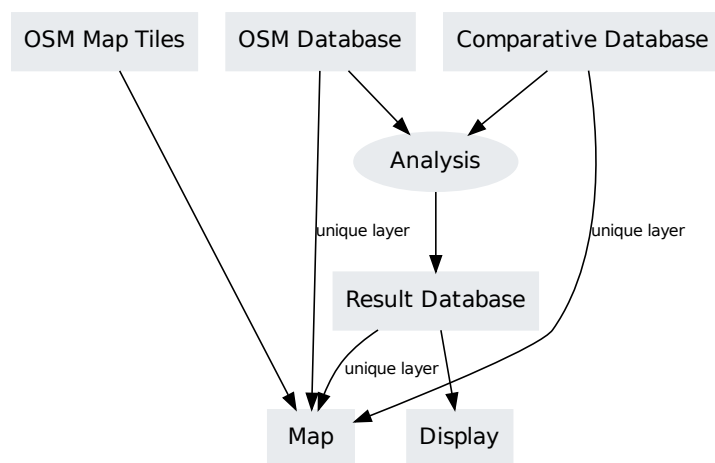


Figure 3.4: The workflow concerning the legal structure

3.4.2 Public Use

If the results are used in any way that is accessible to the public, the restrictions of the licenses become important.

Map-Tiles

The pre-rendered map tiles offered by the OpenStreetMap project are displayed as a background layer in the map presenting the results. This implies the validity of the CC-BY-SA for this result. The comparative dataset as well as the result of the comparative analysis is displayed as a separate layer on top of the OSM-tiles. Since the CC-BY-SA has already been used for several years, a web-page exists that offers help to common license-interpretations [OpenStreetMap Wiki, 2010]. On this page the question “*I created a layer on top of an OSM map. What do I have to put under your license?*” is answered.

The solution presented as “*If you overlay OSM data with your own data created from other sources (for example you going out there with a GPS receiver) and the layers are kept separate and independent, and the OSM layer is unchanged, then you may have created a collective work.*” fits the scenario at hand: ignoring whatever the source of the data is (it does not matter if this data is a part of OpenStreetMap, this will be dealt with later on by the ODbL), the overlay will consist of positions of two datasets, where the location relative to one another and a colour or shape symbol is the interesting information. The tiles in the background are merely there to help estimating where these points are located. The overlays could be presented without any background map and still contain the information important to the viewer.

This way, the map view represents a Collective Work and only the OSM map-tiles are subject to the CC-BY-SA. There is no problem displaying the map in the analysis program. If the user wants to reuse the resulting map it may become a derived work (see OpenStreetMap Wiki [2010] which says “*If you generate a merged work with OSM data and other data (such as a printed map or pdf map) where the non-OSM data can no longer be considered to be separate and independent from the OSM data, it is likely you have created a derivative work.*”) and restrictions apply. This derived work is licensed by the CC-BY-SA and it has to be checked by the user if the displayed data is compatible with that. Therefore, in this second case, the user should be displayed a warning prior to exporting the map. Alternatively he could be required to specify the legal conditions of the comparative dataset. By this, the system can make assumptions about allowed procedures.

Data

There are two different sources of data which will be stored in separate databases: OSM data and comparative data. Since the usage of OSM data by the analysis program can not be seen as “insubstantial” (see section *Insubstantial Use* (page 26)) it has to comply to the ODbL.

The analysis procedure compares the distance of two points of these two datasets. If the distance is close enough, there are two possibilities how this information can be processed:

- a point from the OSM dataset is marked as “included” or “not included”
- a point from the comparative dataset is marked as “included” or “not included”

Either way, the outcome classifies as a derivative database. This assumption is based on [OpenStreetMap Wiki \[2012j\]](#) which says “*If it was intended for the extraction of the original data, then it is a database and not a Produced Work. Otherwise it is a Produced Work.*”. The result of the analysing procedure is defined as the “Produced Work”, the database that was used to create this result or the method and data to recreate it has to be made available. Since the data used to create this database involves the comparative dataset which in case of using data from Tupalo is not compatible with the ODbL, the method and data can not be published. This only leaves the release of the derivative database under the ODbL.

Now it is important to only include information in this derivative database which is allowed to be published. By marking only data of the OSM dataset, the resulting database does not directly contain any information about the comparative dataset and can be released under the ODbL. But instead, when marking or just including points of the comparative dataset, it has to be checked by the user whether or not he is allowed to release this information. This can again be solved by letting the user specify the legal properties of the comparative dataset and then let the analysis program give a recommendation on what the user is allowed to do. The database resulting from the analysis procedure can be published in any case if only Open Street Map features are stored or if the comparative dataset is compatible with the ODbL.

Section 6.1a in the ODbL says that the ODbL does not affect any rights concerning the “[..] *Extraction for purposes of illustration for teaching or scientific research [..]*”. But since the analysis program does not simply extract information this does not apply here.

3.5 Result of this Chapter

3.5.1 Theoretical

By working through the licenses involved, the following conclusions for the analysis program are drawn. The user will have to specify the legal properties of the comparative dataset. By this, the analysis program is able to offer information about what is allowed to do with the output and what is not. Also, there is a “private” mode that allows any operation without limitations or warnings.

The result of the analysis will be stored in a separate database. When not operating in “private mode” this database corresponds to a “derivative database” and its license is the ODbL. Depending on the license of the comparative dataset the program gives information about whether it is allowed to use the results in public or not.

The (non-map) display of the results (e.g. statistics or graphics) are “Produced Works” and can be licensed in any way, if the database is subject to the ODbL. But it is important to only use data from a source compatible with the ODbL. Otherwise the graphics can not be reused.

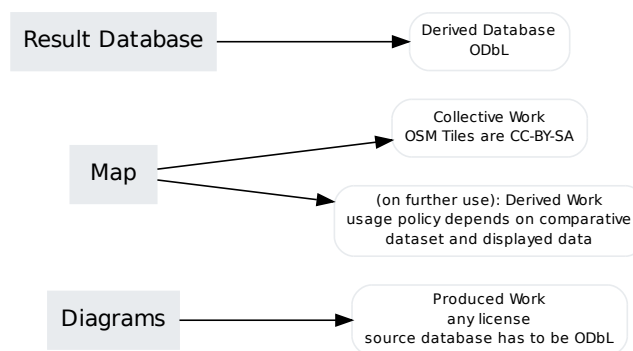


Figure 3.5: Legal circumstances for the analysis procedure (building the result database by marking OSM-objects)

The map is composed of a CC-BY-SA background layer and the data layer which can have any license when only viewed in the application (“Collective Work”). There is also a message warning about further use of the complete map (“Derived Work”) when the license of the comparative dataset does not allow this.

3.5.2 Practical: The “Legal Module”

To integrate the above findings into the web application, a module is programmed that given certain parameters outputs information about what is allowed to do and what is not. The input parameters are given by a JSON document with the following structure:

```
{ privateMode : true;
  comparativeDbOdbLCompatible : false;
  comparativeInResult : false;
}
```

The module offers three functions which each take the parameter-JSON described above as an argument. The return value of each function is either *true* or *false* giving information about what is allowed.

- `allowMapCopy()` Returns true if the displayed map is allowed to be used outside the web page.
- `allowResultDistribution()` Is valid when the resulting data can be exported for external use.
- `allowStatCopy()` Describes whether it is allowed to reuse any images or diagrams generated and displayed by the web application.

The actual processing of information is just a matter of putting together multiple “if” statements. Section *Source Code of the Legal Module* (page 79) in the appendix displays the code of the legal module.

This module represents a starting point for a possibly more complex system that could be used in a more universal manner. For reasons of simplicity there is no distinction made between a dataset that is compatible with the ODbL or with the CC-BY-SA. When the dataset is compatible with the ODbL, it is sufficient.

SIMPLE POI MATCHING

Content of this Chapter

In this chapter a simplified method for comparing two locations is constructed with the help of a geo-spatially enabled CouchDB and tested against a manual check of the same region.

The core of the analysis program is about checking whether two pieces of information of location from different datasets refer to the same point. Hereby the algorithm used for this comparison is limited by legal and technical conditions and makes use of specific features of a geo spatially enabled CouchDB. The matching module is not operable with Internet Explorer since it currently has no support for web workers (see section *Web Workers* (page 19)).

The driving hypothesis for this chapter is the assumption that a simplified matching algorithm will produce acceptable results. To prove this, results generated by the algorithm are compared to a manual check of the same areas.

4.1 Situation

The comparison-algorithm will have the following information at hand:

- one dataset as array of objects
- one database to perform the analysis on
- specific analysis settings entered by the user

The output are objects of the OpenStreetMap database that have a corresponding object in the comparative dataset. Additionally, also objects of the OSM dataset that do not have a partner are included. These results are stored in arrays which are returned. Therefore the algorithm will need a database connection to one database and a way to access the analysis settings. Since the analysis-program is based on a CouchDB (see section *Hypotheses* (page 3)), the database is stored as a CouchDB-database.

This structure makes it possible to keep the matching procedure very modular. When a different method of matching should be implemented, it is possible to exchange this matching module with another one. It is even an option to use a different database system for the objects the check is performed on.

4.1.1 Datasets

The two datasets used for comparison are unmodified data from OpenStreetMap (only operations necessary to store it inside a CouchDB are applied) and a set of points from [Tupalo Internetservices GmbH](#)

[2012].

For the OSM Data, extracts created by Geofabrik GmbH [2012] from 31 August 2012 were used. Tupalo provided two datasets containing all their restaurants in Vienna and Graz represented only by their geographical location and their name. This data is also from 31 August 2012.

4.1.2 Selection of Categories

The comparative dataset will be used as a whole (category wise) which means that all objects contained within it will be used. There are no additional attributes necessary to make a selection.

One has to specify on which selection of OpenStreetMap objects the analysis has to be performed on. The OpenStreetMap database which is prepared for the analysis procedure holds all kinds of objects. Since their tags are also saved alongside the geometrical information, it is possible to filter only wanted entities. According to Gupta et al. [2010] the attributing scheme of OSM data can be described as a “folksonomy”: “A folksonomy [...] is a user-generated classification, emerging through bottom-up consensus.”. Herein Haklay [2010] sees a difficulty: “Ambiguity arises in folksonomies because different users apply different tags to different objects.” When the same type of object from different datasets has to be selected, this poses a challenge. While performing a similar analysis with data from TeleAtlas, Strunck [2010] found out that from 196 (OSM) and 59 (TeleAtlas) possible tags describing POIs only 36 were matching. The best accordance was found with the categories “pharmacy”, “restaurant” and “fuel”.

In this chapter the matching algorithm will be constructed and evaluated by using a dataset containing restaurants. This is because on one hand the available data is limited to this category and on the other hand because restaurants, as stated above, appear to be a category that is more easily matched by classification than others. Furthermore, restaurants represent a category that is easily visible by nature, so they are hard to be overlooked by an OSM-mapper in the field, which guarantees a high value of completeness in comparison to less visible objects.

In the Tupalo dataset, content created by the company itself is mixed with information from other data sources (e.g. Herold [Tupalo Internetservices GmbH, 2009]). So, the homogeneity of positional accuracy is variable but still is accurate enough since the data is intended for orientation by foot.

Data has to be selected from the OpenStreetMap dataset that ontologically corresponds to the comparative dataset. The tag *amenity* provides definitions for “an assortment of community facilities” including restaurants [OpenStreetMap Wiki, 2012k]. On this page, the entries explained in category *sustenance* are *bar*, *bbq*, *biergarten*, *cafe*, *drinking_water*, *fast_food*, *food_court*, *ice_cream*, *pub* and *restaurant*. The tags *restaurant* and *fast_food* cover and do not over-define the representation of *restaurant* in the Tupalo dataset. The other tags stored under *sustenance* better fit to other categories of Tupalo like *Essen und Trinken*.

When using the matching module with the web application later on, the selection of categories is variable. Since there are too many tags to allow a pre-defined selection of all possible combinations of attributes when needed, the user has to be able to define own filters. In the analysis program this is realised by letting the user define own views, as is explained in section *User Selections* (page 54).

4.2 Known Procedures

As mentioned in section *Open Street Map in Research* (page 2) there already exists a number of studies that compare data from OpenStreetMap with other datasets. But nearly all of them are only concerned with the quality of the street-network.

4.2.1 Regarding the Street-Network

- Ather [2009] performs a comparative analysis of main-roads in London. Additionally he does a comparison concerning attributive information but no specific matching of points.
- Ludwig [2010] is also only concerned with the comparison of OpenStreetMap data with the commercial Navteq road-dataset. She found out that the OSM main-roads correlate up to 90 % in German cities, while the correlation drops rapidly when moving to more rural areas or when looking at streets with lower rank.
- Haklay [2010] makes a comparison of only street-network data. (*“Motorways were selected as the objects for comparison [..].”*) His method is not suitable in our context.
- Helbich et al. [2012] perform an analysis of the positional accuracy of road junctions. These are evaluated as points whose distance to one another is measured. Their method is based on the assumption that one already knows which two points of two different datasets are corresponding.
- Chen and Walter [2009] discuss a way to not only compare two street datasets but even improve their quality. Their method may be based on handling nodes of line segments but the actual algorithm is about matching lines. Therefore, their methods concern a different setting than the one needed here.

4.2.2 Regarding POIs

- Schoof [2010] performs a positional analysis of mostly street-network data (*“Der Schwerpunkt der Vollständigkeitsanalyse liegt auf den linienhaften und flächenhaften Objekten.”*). She does a simple quantitative comparison of POIs on page 34. But because of the simplicity and incompatibility of the comparative dataset concerning POIs this can not serve as an example in this case.
- Cipeluch et al. [2010] perform a comparative analysis of the street-network as well as of specific points-of-interest. They compared data from OpenStreetMap with GoogleMaps and BingMaps. Since they could not get hold of vector data from either of the two commercial datasets, their method is a visual comparison. In this thesis this procedure is only applied to evaluate the result of the automated matching algorithm.
- Strunck [2010] performs an analysis about POIs in Germany. He is comparing selected categories of POIs of the OSM dataset with data from TeleAtlas. Hereby he is checking **positional accuracy**, the **spatial distribution** of completeness and the **growth** of data. While neither the growth nor the spatial distribution of completeness is relevant for this thesis, his findings concerning the positional accuracy are of interest.

4.3 Acceptable Results

As specified in the research hypothesis, the algorithm has to deliver “acceptable results”. To define a measure for this, a list of conditions is formulated that have to be met. It is important to point out that we are not interested in the actual result of the checking procedure but in how well the algorithm operates (e.g. how many false/true positives/negatives exist).

4.3.1 Accuracy of POIs

Strunck [2010] performed an analysis with POIs of TeleAtlas in Germany. He made a buffer-search with 5, 10 and 20 meter-buffers and compared found POIs of a selected category. He only compared points that have been identified as having a corresponding partner by their name. This means, that the resulting value not only gives information about the completeness of the dataset but also about the efficiency of the method used. The result is that within a search-radius of 20m, only 5.78% of tested POIs concurred. When looking at objects with smaller extent the results get better.

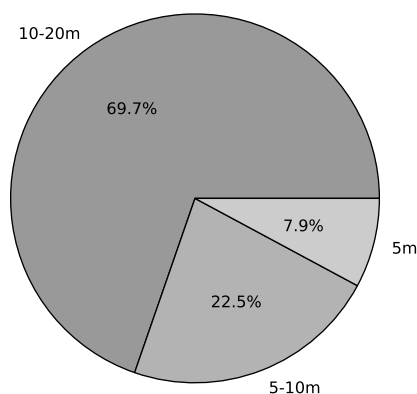


Figure 4.1: Distance of POI between OSM and TeleAtlas in percent
(source: see [Strunck, 2010], fig.5)

From these 5.78%, most of the POIs were situated between 10m and 20m, as can be seen in *figure 4.1*. Unfortunately Strunck did not use bigger radii for comparison. Strunck holds the big extent (e.g. airports) and the variable positioning some of these points responsible for this result (see Strunck [2010], p75).

4.3.2 Error Matrix

A method to analyse the accuracy of classifications named an error matrix was investigated. It is described by Oczipka [2007] in chapter 5.6 who in turn took his methods from Congalton and Green [1999]. The error matrix is a way to deal with remotely-sensed information but also seem applicable in case of this matching procedure. Congalton and Green [1999] say on page 75 “An error matrix is a very effective way to represent map accuracy [...]”. As Oczipka describes, the error matrix is a table where the columns represent categories of classification from a reference dataset (in our case the manual matching) and the rows stand for the same categories of the automated matching. For this thesis, the error matrix could be organised as layed out in *figure 4.2* (page 35).

Despite its first impression, the error matrix proved not to be suitable for our case. While *true positives*, *true negatives* and *false negatives* can be identified easily, it is problematic with *false positives*. A *false positive* occurs when a node is matched to another one but should not be at all. But this classification is also valid when a point should be matched to another one but is paired with the wrong partner. This makes the definition of *false positive* unclear and prevents from a straightforward interpretation of the results of the error matrix. Therefore an error matrix is not used.

		manual matching			
		✓	X	Σ	U(a)
matching algorithm	✓	true positives	false negatives	row sum	U(a) hit
	X	false positives	true negatives	row sum	U(a) miss
	Σ	column sum	column sum	total sum	
	P(a)	P(a) hit	P(a) miss		
O(a)		overall accuracy			

Figure 4.2: Layout of the error matrix to evaluate the matching procedure
 (P(a)) Producers Accuracy, (U(a)) Users Accuracy, (O(a)) Overall Accuracy, source: see Oczipka [2007], tab.10 (modified)

4.3.3 Measures

When comparing the manual check with the automated one three different categories are used to rate the results which are also shown in *figure 4.3*.

- **Auto Match Right:** This includes every correct result, no matter whether the node has to have a match or not.
- **Auto Match Overlooked Partner:** This means that the automated matching procedure classifies this point as single, but according to the manual matching, in reality it has a corresponding partner that was ignored.
- **Auto Match Wrong Partner:** In this case a point was matched but it was the wrong one.

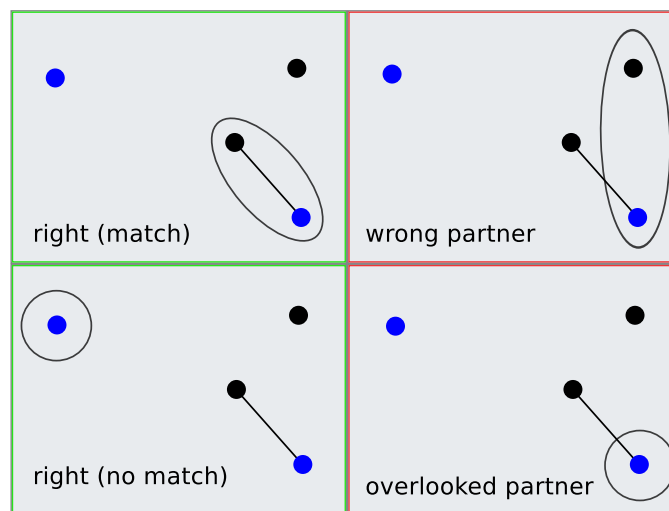


Figure 4.3: Different cases of matching differentiated
 (blue) OSM-POI, (line) pairs as they should be, (ellipse) result of the algorithm

Based on this measures certain ratios are calculated:

- **hit-rate** describes the percentage of points that were matched correctly in contrast to all wrong results.

- the **miss-rate** is an equivalent value to the hit-rate describing the percentage of points that were not matched correctly.
- **Percentage values of two error types** are calculated as well. They describe when a possible match was “overlooked” or when a match was performed with the “wrong partner”.

These values give a general idea of how well the algorithm performs under different conditions (e.g. search radius and location).

4.3.4 Criteria for an “accurate enough” Matching-Algorithm

It is not possible to give a general definition for “accurate enough” since the optimisation of the different measurement values depends on the further use of the result. If one is doing a e.g. quantitative analysis, a few wrong results compared to hundreds of right ones will not make a difference. But if the user wants to exactly match multiple POIs, not even one false match may be allowed. The criteria named here concentrate on the first case where a certain amount of errors is tolerable.

If one assumes that there is not much difference between the density and quality of POIs between Germany and Austria, Strunck's value (5.78%) represents a desired minimum hit-ratio when using a search-radius of 20 meters. Actually this value is not quite satisfactory for practical use but it points out that this matching algorithm is producing similar results as the study performed by Strunck [2010]. Because the matching-algorithm used in this thesis operates on geographical coordinates, the radius of 20 meters needs to be transformed. For the area of Austria a value of 0.00045 degrees (0° 0' 1.62'') loosely corresponds to an extent of 16 x 25 meters which seems appropriate.

For the algorithm to be of real use, it has to reach a matching rate of at least 2/3 (66.6%) when using an optimal search radius. This shows that there are measurably more correct results than wrong ones.

Because of the extended range, a bigger search radius also increases the risk of false-positives which stands in contrast to the targeted accuracy. While this problem can not be solved easily, a limitation can be applied: One point from the comparative dataset is only allowed to be matched one time. Due to the simplicity of the algorithm it is not possible to determine which of possible multiple pairs is closer to each other and thus qualifies as the correct one (one would need a two-pass-approach to achieve this, see section *Two Passes* (page 44)).

It is expected that a simple bounding-box search delivers equally good results as a radius-search. So, the hit-rate has to be the same with both methods. By “equally”, a maximum difference of 2 matches is allowed. This is an argument for the simplicity of the algorithm.

Taking the results of the previous studies into account, the developed matching algorithm can be treated as acceptable when it is able to generate results that fulfil the following criteria:

- The **hit rate has to be at least 6% when using a search radius of 0.00045°**.
- The **hit rate should reach 66.6%** with the best search radius to ensure a minimum of quality.
- One point is allowed to only **be matched once**.
- A search by a **simple bounding-box makes no difference** in comparison to using a radius search.

The algorithm will be checked against this criteria in section *Evaluating the Method* (page 45).

It is important to mention that depending on the further use of the results of the matching procedure one may need different values of accuracy than the ones used here.

4.4 Developed Procedure

Ongoing from already known procedures (see section *Known Procedures* (page 32)) a matching algorithm is developed that is operating on a CouchDB installation with GeoCouch extension. The aim is to create an algorithm as simple as possible that still delivers acceptable results. There are ways for further improvement which are discussed in section *Further Optimisation* (page 43).

4.4.1 Limitations

There are certain limitations the matching algorithm is exposed to.

Legal Limitations

As defined in section *Result of this Chapter* (page 29) certain legal restrictions apply. The user specifies the legal property of the comparative dataset. The matching-procedure itself can be protected from making most decisions on how to operate by restricting it to blindly work with two sets of input-data. Therefore, the matching procedure only has minimum access to the databases and is getting and sending data by the use of GeoJSON, as described in section *Interface - GeoJSON* (page 22).

The output of the matching procedure may contain more than just the objects of the OSM database. It is up to the web application to remove results when they are not compatible with the ODbL.

Technical Limitations

Due to the nature of spatial queries currently possible with GeoCouch, one can only make either a query optimised for location or for attributes. As will be explained in section *The Simplified Matching Procedure* (page 37), the check for position is done by iterating through a given array of POIs and performing a spatial request on a database for each point. The result will be an array of all point objects that have a corresponding POI in the comparative dataset.

This means that the algorithm needs enough information to make queries on a database. This can be achieved by providing the matching-algorithm with the URI used to perform a spatial query.

To speed up the checking procedure, “web workers” are used (see section *Web Workers* (page 19)). One limitation of this technique is, that one is not allowed to access the DOM of the Website and therefore cannot load any jQuery modules which in turn makes it necessary to implement special functions to access the database.

The memory provided by the computer system on which the matching procedure is running on may define the extent possible to check. While the supposedly most memory intense process (checking the location) is done by a database-query (the spatial request), all objects that have to be compared from the OSM-database and the results are kept in-memory to iterate over during the check.

4.4.2 The Simplified Matching Procedure

The term “simplified matching procedure” refers to the algorithm used. “Simplified” in this case means the following:

- only one pass through available data
- only use positional information to perform the match

There are two ways to perform a simplified check for location by the means possible with this system. Either take a list of points from the comparative-dataset and perform a positional check for each of them on the OSM-dataset or vice-versa. One may say that this does not really concern the internals of the checking procedure because it can operate on any dataset but the role of the OSM dataset makes a difference to the way it is handled and prepared.

In the following two sections the two possible methods are presented.

Check on OSM (1)

The first method (see *figure 4.4*) takes the comparative-dataset and performs the check on the OSM-database.

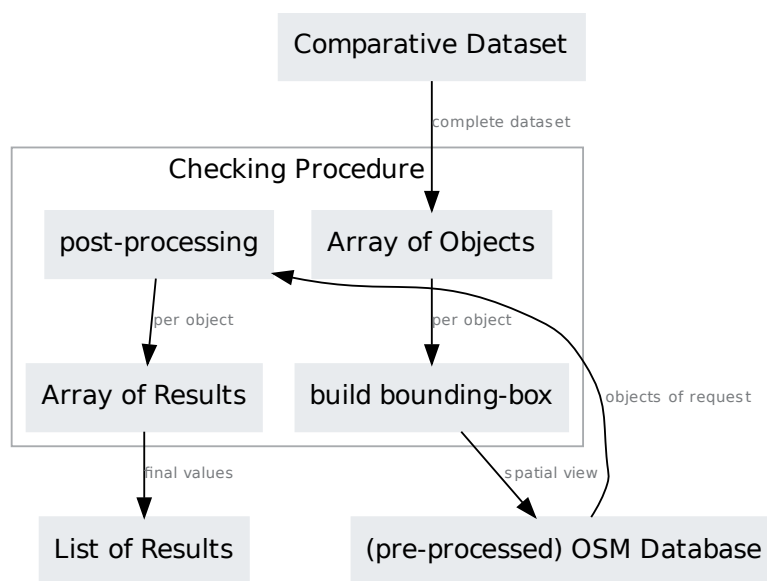


Figure 4.4: The checking procedure on the OSM database

The good thing about this method is that there is no need to save the comparative dataset in a database of its own since the matching algorithm just needs an **array of input objects**. This array already defines the geographical extent of the area that has to be checked. After receiving these objects, they are operated on one by one. For each entity a **bounding-box** is calculated and a spatial view which also filters for certain categories is executed on the OSM-Database. The extent of the bounding-box defines the search-distance. Any value retrieved may have to be **post processed**. This is necessary since only one point out of the result set can be a match.

Also, the **OSM Database** has to be pre processed because POIs are not only stored as nodes, but sometimes are assigned to line-strings that compose an area (e.g. buildings). Therefore, an abstracted version of the OSM-Database is needed which may pose obstacles when future optimisations are applied to the system.

All final **results** are collected in an array which then is returned by the matching-procedure.

Check on Comparative-Dataset (2)

The second method (see *figure 4.5*) takes an array of OSM-objects and performs the check on the Comparative Datasets.

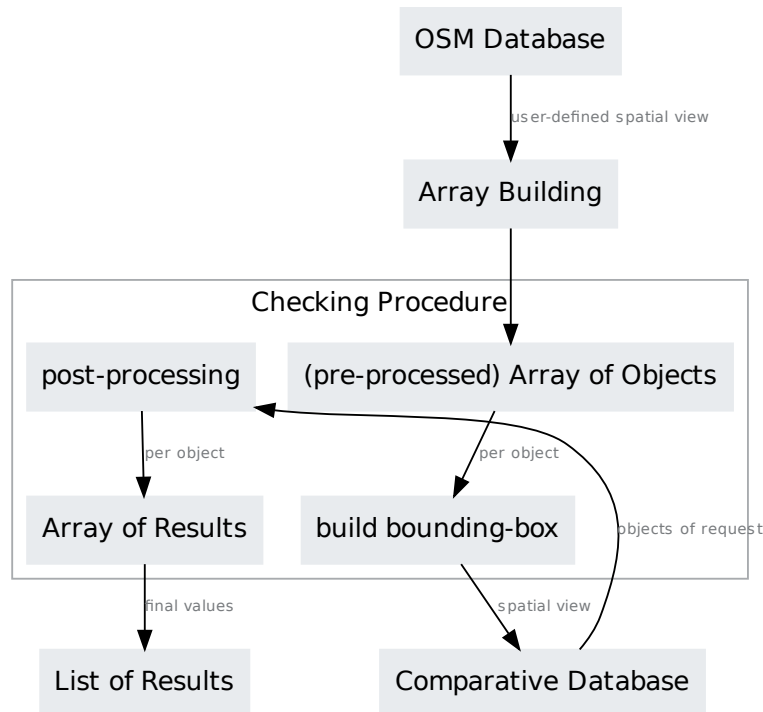


Figure 4.5: The checking procedure on the comparative dataset

This algorithm gets an **array of OSM-objects** as input, which defines the extent (for the algorithm) and type of data the check will be performed on. While using the first method, this array of objects also represents the extent and selection of objects that have to be checked for, it is different for the second one: usually one can assume that the OSM-Database consists of more elements to a greater extent than the comparative dataset. Thus, a selection has to be made before sending the array of OSM-objects to the algorithm as is described in section *User Selections* (page 54).

A big advantage of this method is that the **OSM-database** does not need to be pre processed, since any transformation necessary can be done while iterating over each object during run time or at least after the query on the OSM-database.

The further procedure is consistent with the one of method one (section *Check on OSM (1)* (page 38)).

Selected Method

Each of the described methods has its advantages. In the present case it was decided to go on with the second method, because it can operate on a non pre processed OSM database which in turn lets the complete system stay flexible. The selection of region and category can be done before sending the OSM-data to the algorithm by a user built spatial view. The downside is that it is absolutely necessary to import the comparative dataset into a database for the CouchDB to execute spatial views on it. But since there is no limit on the amount of databases that can be used, this is only a minor problem.

4.4.3 Building a Usable Module

As described in section *Modular Programming* (page 21), the analysis program will make use of independent modules. The comparison of two POIs is one of these. The programming language used is JavaScript. The module itself will consist of a folder and a *.js file with the same name. In this case the module is named *mod_m_simple*. Each module has the prefix *mod* and the matching algorithm has an additional one named *m*.

The interfaces and components of the complete matching-module are displayed in *figure 4.6*.

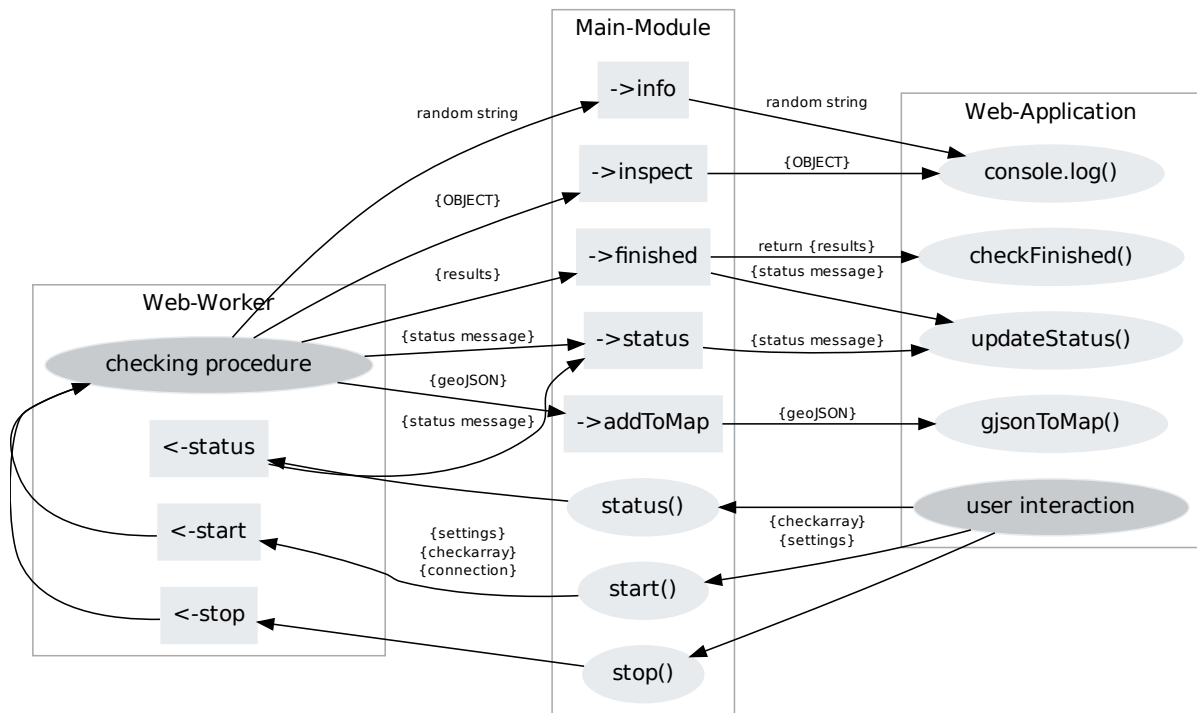


Figure 4.6: The interfaces and components of the matching module (functions) are displayed by an ellipse, (postMessages) are displayed by a box

The Module Interfaces

Communication is necessary for the web application (external interface) and between the main module and the processing web worker that is part of the matching-module (internal interface). Also, there need to be three user defined functions present in the web application which will be called by the matching algorithm.

External Interface

The interface of the module offers one method to invoke the checking procedure, one to request the status of the procedure and one to stop the procedure premature.

Functions of the Module

- *start()* needs two parameters: an array of geoJSON to iterate over (module-internally called *checkarray*) and settings (module-internally called *settings*).

The array of GeoJSON objects has to be prepared by the hosting web application (see section *Categories* (page 54)) and is actually a valid GeoJSON “Feature Collection” or a single “Point” or “Polygon” GeoJSON object as it is described by Butler et al. [2008] in section 2.3.

The settings are also passed on in the form of a JSON-object that at least contains the following structure (the values represent default settings):

```
{
  connectionstring : '',
  searchradius : 0.000225,
  radiussearch : false
}
```

By this it is possible to deliver a varying amount of information to the matching module without the need to adjust it in order to cope with more information than necessary, as long as all the information needed by the module is included. `connectionstring` holds the URI of the database connection. In our case this is the complete URI to the CouchDB database and the spatial query. Additionally to that, one has to specify `searchradius` which holds the value defining the extent of the window where the algorithm is looking for matching points in decimal degrees and `radiussearch` which gives information about whether to use a simple bounding box search or using an actual circle.

- `stop()` simply halts and terminates the web worker and exits the matching module
- `status()` triggers a status update by calling a function provided by the web application.

Functions of the Web-Application

The web application itself has to provide the functions `gjsonToMap()`, `updateStatus()` and `checkFinished()`.

- `gjsonToMap()` needs a GeoJSON as parameter. This function should serve as a means to display objects while the matching procedure is running. Of course, depending on its implementation, this function can operate on the delivered GeoJSON object in any way it is necessary. The GeoJSON object contains additional entries named: `layer` which describes whether it is a hit or a miss (or none of both), the field `checkdistance` which contains the distance to the point this object was matched to and `partnerId` which holds the document id of the database entry of the matching partner.
- `updateStatus()` gets a JSON object in a specific format as parameter which looks like this:

```
{ title : "Matching ...",
  status : 29
}
```

`title` is a field giving the name of the process and `status` can be any information - it is recommended to fill it with a percentaged value.

- `checkFinished()` is called by the module when it has finished processing. This function receives all the results in form of a JSON document with the following structure:

```
{ hit : [geoJSON],
  miss: [geoJSON],
  hitc: [geoJSON]
}
```

All information about the objects is provided by their GeoJSON representation. `hit` is a list of all objects from the OSM dataset that were matched, `miss` contains all objects

with no partner and `hits` is a list with all matched objects from the comparative dataset. Any information about the amount of matched objects can be gathered from the length of these arrays. Also, the `checkdistance` and `partnerId` (as explained above) are also included in each object.

Internal Interface

The communication between the main module and the web worker is only possible by strings (as was explained in section *Web Workers* (page 19)) that are represented by JSON in our case. As decided by the author, this communication-JSON holds an entry named `command` and one called `content`:

```
{  command : 'status' ,
  content  : '',
  db       : ''
}
```

Depending on the actual `command`, additional keys and values like a database connection in the key `db` may be included.

Communication to the Web Worker

The value of the key `command` of the JSON string used to communicate can be:

- `start`: needs `content` to hold the array of objects to perform the matching procedure on
- `stop`: does not need any `content`
- `status`: does not need any `content`

Communication from the Web Worker

The worker can send the following `command` tags to the main module:

- `info`: lets the web application display debugging information in the JavaScript console since the web worker has no access to it
- `inspect`: makes the same as `info` but transfers an object for inspection to the JavaScript console
- `addToMap`: triggers the `gjsonToMap()` function
- `finished`: transfers the results to the main-module and terminates the web worker
- `status`: triggers the `updateStatus()` function

The Main Module

The listing in section *Source Code of the Matching Module* (page 80) in the appendix displays the source code of the main module. The main module primarily serves as module interface and agent between the web application and the web worker.

The lines 26, 28, 31, 43 and 46 define callbacks that are executed when the web worker posts a certain message to the main module. When the module is initiated with the method `start()`, the `checkarray` is transformed to an array if it was a single geoJSON object and the web worker is started with the necessary parameters.

The module may receive commands from the web worker which are caught by the `worker.onmessage` event in line 25. The most important one is at line 31

`message.data.comment == "finished"` which is executed when the matching procedure has finished. The module generates a final status message and calls the `checkFinished()` function of the hosting web application which also delivers the results of the matching procedure.

The Web Worker

The code of the web worker can be viewed in the appendix in section *Source Code of the Matching Module* (page 80). The web worker is the core of the matching algorithm and handles the actual comparison. When it is initialized it starts the matching procedure. After finishing the procedure the complete set of results is communicated with the main module.

The general architecture of the web worker is as follows: First, the `start` command is received (line 12) and for each entry in the delivered array of objects that have to be checked a search is performed:

- After building an individual search query (line 33) it is executed and its results are cleared from already checked points which should not be matched again (line 38).
- Then the distance of all points left over to the one the check is performed on is calculated and only the nearest one is kept (line 42).
- If a radius search is specified the remaining point will also be checked against that (line 45).
- The following code is about adding the point to the list of results and drawing the search area and the line connecting the two matching points on the map.

When the iteration over all points to be checked has finished, the module is informed and the results posted to the module (line 83). After this the worker terminates itself (line 88).

It should be mentioned here that a radius search is only performed on points. When dealing with buildings a bounding box request is used. This does not pose a problem because buildings are not encountered often and the results did not vary greatly when switching from a radius search to a bounding-box search (see section *Discussion of the Criteria* (page 47)).

4.5 Further Optimisation

By using methods described in this section it is possible to extend the simplified matching algorithm.

4.5.1 Name Matching

If an attributive information is given additionally to the geographical one, a matching by name can be performed. This name matching can be of use in three situations:

- by only using points that have the same name from the beginning
- post processing the result after the actual matching procedure
- using name matching as part of the algorithm (e.g. filter multiple possible results)

Ludwig [2010] names two methods to determine if two strings of words refer to the same object on page 58. The first one is a kind of pre processing, namely to disassemble a string into single “tokens” and then comparing these. The second one is the “Levenshtein” distance which describes the number of modifications necessary to make the two strings equal.

Of course, any attribute that is present in both datasets can be used to perform a comparison.

4.5.2 Local Positional Accuracy

If it is possible to determine a value for positional correctness for the area around each point, the checking radius can be adapted to it.

Ather [2009] formulates the hypothesis that areas with fewer named roads should have a lower positional correctness since fewer users are editing these regions. This proves true as he describes on page 62. But he also points out that this is a local measurement and is based only upon a selection of roads. Further analysis reduced this measurement to a “*slight positive trend between number of users and road name attribute completeness*” (p.68). Hence, the “*number of users did not appear to have any effect on the positional accuracy of the dataset.*”.

4.5.3 Statistical Evaluation

While proposing a method for comparing line features, Goodchild and Hunter [1997] mention that the value of distance measured when comparing point features provides a good basis for statistical analysis.

Performing these on the results and integrating them into a possible “Local Positional Accuracy” property may improve the matching quality.

4.5.4 Two Passes

By using just one pass there is a certain amount of points that despite being the nearest ones of the result set of a bounding-box request they are still not the nearest pair possible.

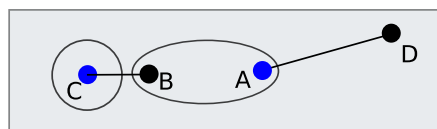


Figure 4.7: Example situation where a two pass method would work better (blue points) OSM-nodes, (black points) entries from a comparative dataset, (lines) matches like they should be, (ellipse and circle) the calculated, wrong results)

Figure 4.7 shows a situation like this. The points B and C as well as A and D are pairs but the algorithm as it is would match B and A together and identify point C as a node with no match. Since point A is checked first, the resulting list of possible matches for point A would be the points B and D. B is nearer than D, so a connection is made between B and A and point B is marked as “already matched”. Now, point C is checked. There are no possible results because point B was already matched even when B is nearer to C than to A and point D is too far away.

With a two pass approach, first the nearest possible matching partner for each of the comparative points (black) could be determined. When performing the actual matching afterwards it is possible to only use points from the result set that are not nearer to any other points they could be matched to. In our example, when making a check with A, point B would be ignored since it is nearer to C than to A. D is the only possible partner left for A.

4.5.5 Manual Correction

The possibility to define certain matches manually before performing the actual automated matching would be a way to avoid consecutive faults as they are described in section *Common Errors of the*

Algorithm (page 48) to a certain degree.

4.6 Evaluating the Method

To test the result of the developed algorithm, a manual analysis of the checked region is done and the results are compared to the ones of the matching algorithm. Furthermore the criteria defined in section *Acceptable Results* (page 33) have to be met to classify the algorithm and its results as “acceptable”.

Four regions have been selected to perform the comparison on, namely the city districts “Josefstadt” and “Wieden” in Vienna and the city districts “Lend” and “Innere Stadt” in Graz. After specifying the ideal matching situation by a manual analysis (reference), each of these regions is analysed four times, one time with a search radius of 0.00045° , one time with 0.0009° , one time with 0.000225° and one time with 0.0001125° .

4.6.1 Manual Analysis

To create a reference to which the automated matching procedure can be compared to, a manual check is performed. This is done by using a map displaying both POI sets together with the names of their POIs. As an additional help, buildings from the OSM database are visualised in the background. Thus it was possible to identify all possible pairs of points. *Figure 4.8* shows an excerpt from one of these maps.

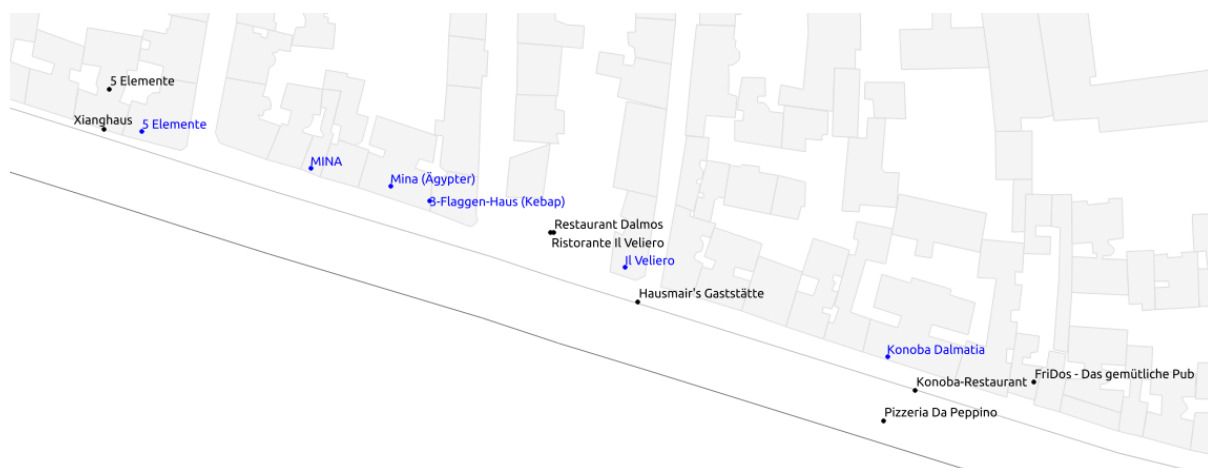


Figure 4.8: An excerpt of the map used for manual matching in the area of Josefstadt
(sources: Geodaten © OpenStreetMap und Mitwirkende, CC BY-SA, Tupalo 2012)

4.6.2 Matching Algorithm vs. Manual Analysis

The results of the comparison between the two matching methods are shown in the maps in section *Comparison Maps* (page 88) in the appendix.

The diagrams in *figure 4.9* (page 46) give an overview over the performance of the automated matching algorithm and the amount and type of errors that occurred.

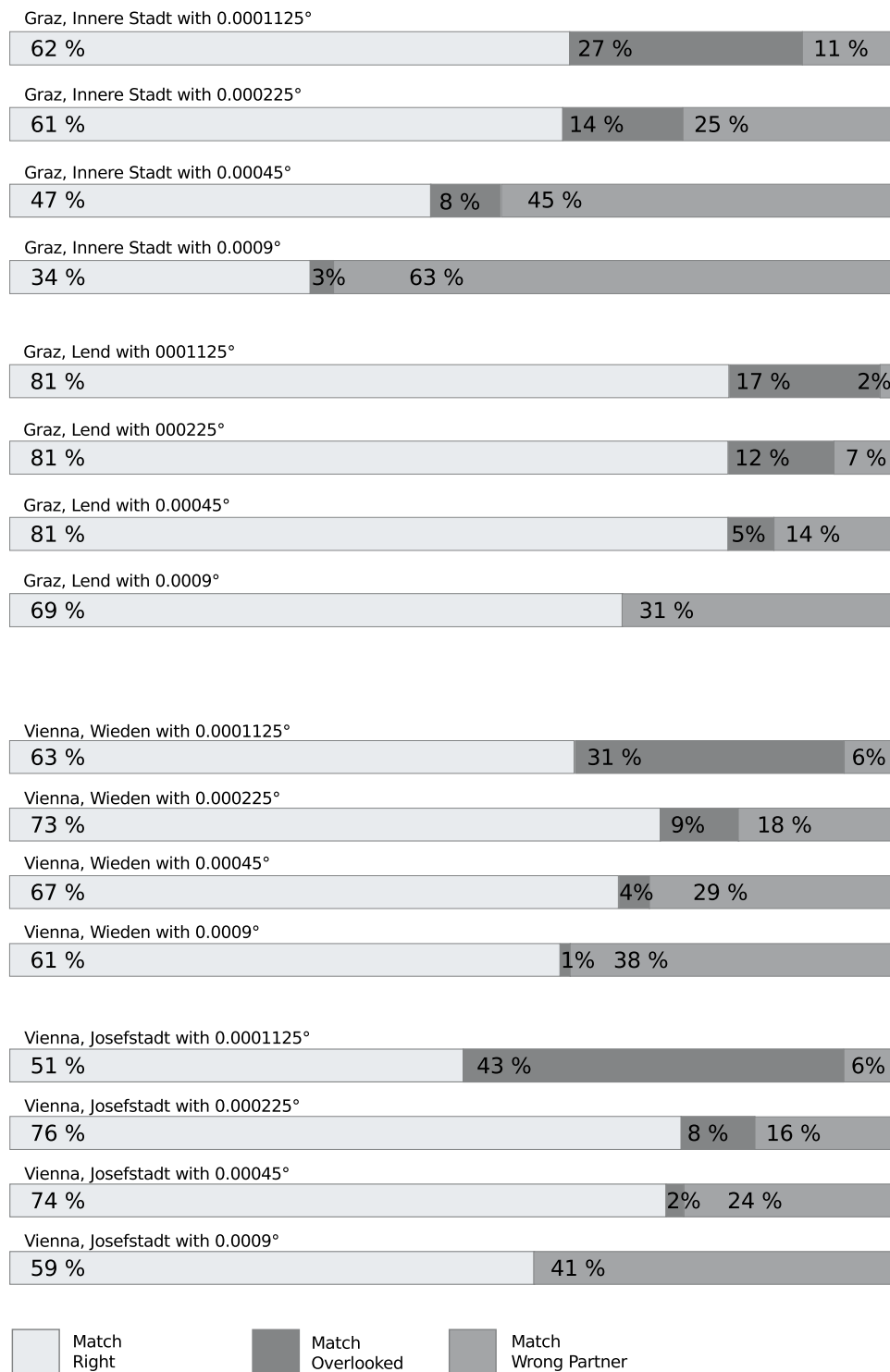


Figure 4.9: Results of comparing the automated and manual matching

4.6.3 Discussion of the Criteria

The results of the comparison of matching procedures are checked against the criteria formulated in *Criteria for an “accurate enough” Matching-Algorithm* (page 36) to determine the usability of the algorithm.

The hit rate has to be at least 6% when using a search radius of 0.00045°.

When using a search radius of 0.00045° the hit-rate lies above the targeted 6% in all cases, so this criteria is fulfilled.

Criteria One ✓

The hit rate should reach at least 66.6% with the best search radius to ensure a minimum of quality.

With the exception of “Innere Stadt” in Graz, all areas produce results better than the expected 66.6% hit-rate with their best search-radii. The bad result for this area may derive from a higher density of points. *Figure 4.10* (page 48) displays the average of all minimum distances for each point in its area. Interestingly the average for “Innere Stadt” is in fact the smallest one. Another reason could be a general difference in mapping accuracy. The area of “Innere Stadt” is located in a more hilly spot which may make it more difficult for mappers to rely on their GPS tracks or potentially not well corrected orthophotos. When making a visual comparison (points mapped on the street vs. points mapped inside building outlines), there does not seem to be a great difference in the localisation of the points. So the second criteria to reach 66.6% with the best search radius can not be met for all areas at this time.

Criteria Two X

One point is allowed to only be matched once.

The algorithm is constructed in such a way that each point is only matched once. This is true for the data that is checked as well as for the comparative dataset. Therefore, the third criteria, which requires exactly this to happen, is met. If necessary one would be able to deactivate this feature easily but this does not offer any significant advantage as there is no user friendly way implemented to do it.

Criteria Three ✓

A search by a simple bounding-box makes no difference in comparison to using a radius search.

Nearly all tested areas and all tested radii produce the same results when switching to a simple bounding-box query instead of a radius search. This may be because when there are multiple results the algorithm always takes the points nearest to each other as pairs. The only cases when there is a difference of one match is within the area of “Innere Stadt” in Graz when using a search radius of 0.000225° and “Wieden” in Vienna with a search radius of 0.0001125°. This criteria is fulfilled.

Criteria Four ✓

While not all required criteria are met to classify the matching algorithm as “accurate enough” it may still be suitable for special cases. It seems to work well enough especially in areas with a low to medium point density (see *figure 4.10* (page 48)). Also, the system is constructed in such a way that it is possible to modify or extend the existing code to enhance the method to deliver better results.

One can see that in nearly all areas the biggest search radius (0.0009°) produces the worst results while only a few points are overlooked. When shrinking the radius, points that belong together are ignored but

Lend	0.00131°
Josefstadt	0.00083°
Wieden	0.00066°
Innere Stadt	0.00044°

Figure 4.10: Average minimum distances of points in the checked areas

the general miss-rate is reduced too. The gain of the hit-ratio is quite remarkable between 0.0009° and 0.00045° in all areas while it does not change that consistently between 0.00045° and 0.000225°. Lend and Josefstadt show only a marginal or no better hit-ratio. When looking at the difference between the radii of 0.000225° and 0.0001125° one can see two cases: first, that there is no change of the hit-rate but a shift of the error-rate from “wrong partner” to “overlooked” (both areas in Graz) and second a gain of the miss-rate in favour of “overlooked” matches (both areas in Vienna). An increase of “overlooked” errors between these two radii can be seen in all areas which seems plausible since a smaller radius excludes more POIs from each check.

As there are nearly no overlooked matches in all four areas with the largest search radius, this radius is almost equal to using no search radius at all.

As expected, when the search radius is getting smaller the amount of “wrong partner” errors decrease but the “overlooked” errors increase. Since an “overlooked” error is easier to deal with afterwards (there is no partner left if the single match is reverted manually) it should be preferred over a “wrong partner” error. According to [Cobb et al. \[1998\]](#) such an “overlooked” error corresponds to the type “false negative” which is the kind of error with the least consequence.

4.6.4 Common Errors of the Algorithm

To better understand the results some common errors the algorithm can not compensate for are described in this section.

- **Consecutive Faults:** If a point is matched to the wrong POI, but this very POI would have been matched correctly if it would not have been already matched to the previous point, the next check has to be wrong even when this POI is the most plausible one to the algorithm (since it is not allowed to match any point two times). This kind of error can clearly be seen in the comparison map of Graz “Innere Stadt” with 0.0009° (see [figure 7.16](#) (page 104)) but occurs quite often with this search radius in general.
- **Errors in the dataset:** During the preparation and execution of the matching procedure, it became clear that both datasets that were used contain errors like POIs that are double-mapped or errors in the naming. While this is to be expected from every dataset of a certain size the algorithm can not compensate for these errors and produces wrong results when encountering them.

4.7 Result of this Chapter

In this chapter a simplified matching algorithm was designed and evaluated. The criteria defined in section [Criteria for an “accurate enough” Matching-Algorithm](#) (page 36) could not all be met even though it was a close call. The algorithm is still usable especially because it only failed with one area (“Innere Stadt”, Graz). The average hit-rate of the best results of all four areas is 73% which would be

sufficient. Possibly, one could use the average minimum distance of all points as a measure to determine whether this algorithm is suitable or not. The average minimum distance could also be an index for specifying the optimal search radius for a certain region. This requires further research.

The algorithm performs best with search radii between 0.000225° and 0.0001125° with the dataset used. For the best averaged result, one should use a radius of 0.000225° or similar and rather reduce than expand it.

A simplified matching algorithm like the one used in this chapter is not sufficient for an approach where each single match counts. It can be of use when trying to detect tendencies or majorities of any kind.

What has not been determined is how well the algorithm performs when using other categories or if the area is not inside a city.

INTEGRATING THE SYSTEM

Content of this Chapter

The matching module is integrated into a user friendly web application. To fulfil its functionality only a geospatially enabled CouchDB and a web browser capable of HTML5 is necessary.

The matching algorithm proposed in the previous chapter needs user friendly means to use it. As defined by the hypothesis, the complete system has to be hosted only by a CouchDB and work with large datasets. An HTML5 web application is created and its functionality is tested.

This chapter is very practical and shows one possible way of compiling a web application using the matching module. The source code of this complete application can be found at <https://github.com/scubbx/simplematcher> or on the SD card located at the end of this thesis. This SD card also contains a VirtualBox image with a running version of the matching program pre loaded with OSM data of Austria and the cadastre of trees of the city of Vienna [Wien, 2012] as a comparative dataset. To lower the amount of disk space necessary, roads and nodes with no attributive information at all have been removed from the dataset used on the SD card.

5.1 Circumstances

The creation of the web application happens with certain expectations which are clarified in this section.

5.1.1 Requirements of the System

The web application has to act as a connector between the matching module and the end user. It also serves as a blueprint for possible further applications. The user has to be able to specify the parameters of the matching procedure. The results of the match have to be displayed and the possibility to download the results in a machine readable format has to be provided. The results of the legal analysis made in chapter 3 also have to be integrated.

5.1.2 Tasks to be Fulfilled

To verify the hypothesis of this chapter (*The complete system can be contained within one independent, single database and perform on large-scale analysis in an efficient manner.*) the following steps have to be taken.

- create a web application that integrates the matching algorithm (chapter 4) and the legal module (chapter 3) and is hosted by a single CouchDB database
- replicate the final system to another database and verify that the system is functional
- test the system up to national (Austria) extent

When all of these points can be met, the system is considered functional and the hypothesis defining this chapter is verified.

5.2 Implementing the System

To create a web application supporting the matching algorithm, one needs to prepare two things: a database storing OSM data and the HTML5 application itself. In this section the implementation of both of these is explained. In the end, the web application and the OSM dataset are stored in the same database.

5.2.1 Preparing the Database

For the database the default installation of Apache CouchDB version 1.2.0 is used [Apache Software Foundation, 2012]. Additionally to that the experimental branch of the GeoCouch extension [Mische, 2012] which allows queries not only by bounding boxes but random polygons is compiled and installed.

Although CouchDB and GeoCouch are not as established as traditional database systems, there is already more than one way to populate a CouchDB database with OSM data. At the time of writing this thesis there are three ways known to the author:

- a combination of programs and scripts called “OSMCouch” developed by emka [2012]
- a patched version of “Imposm” [Tonnhofer, 2012]
- the tool “ogr2ogr” from the GDAL suite [GDAL Development Team, 2012a] in recent versions offers a CouchDB driver [GDAL Development Team, 2012b] to directly write geodata to CouchDB databases [Apache Software Foundation, 2012]

The author decided to use the first method because it is easy to understand and to control. It consists of two parts:

- A configuration file for a program called *osmjs* [Topf, 2012b] which allows filtering and processing raw OSM data. Its output is a JSON document containing an array of GeoJSON entries, each representing an OSM entity.
- A python script that takes the generated JSON file as input and uploads its content to a specified CouchDB database.

The source OSM file used is the extract of Austria from 18 October 2012 downloaded from Geofabrik GmbH [2012]. The resulting database holds more than 24.000.000 documents. The import generates “Point” and “LineString” features only. When excluding never used elements before importing the dataset into a CouchDB database (like nodes with no attributes at all or highways) the number of documents generated was reduced to about 2.000.000 entries. For the integration of the system the complete dataset was used.

5.2.2 Preparing the Web Application

The web application is implemented as a CouchApp like described in section *CouchApp* (page 17). It is saved to the same database as the OpenStreetMap dataset. Not all features offered by a CouchApp like “evently widgets” or “mustache templating” are used. While it would have been possible, it was not necessary. They would have made the source code more difficult to read for someone who has no experience with these functions.

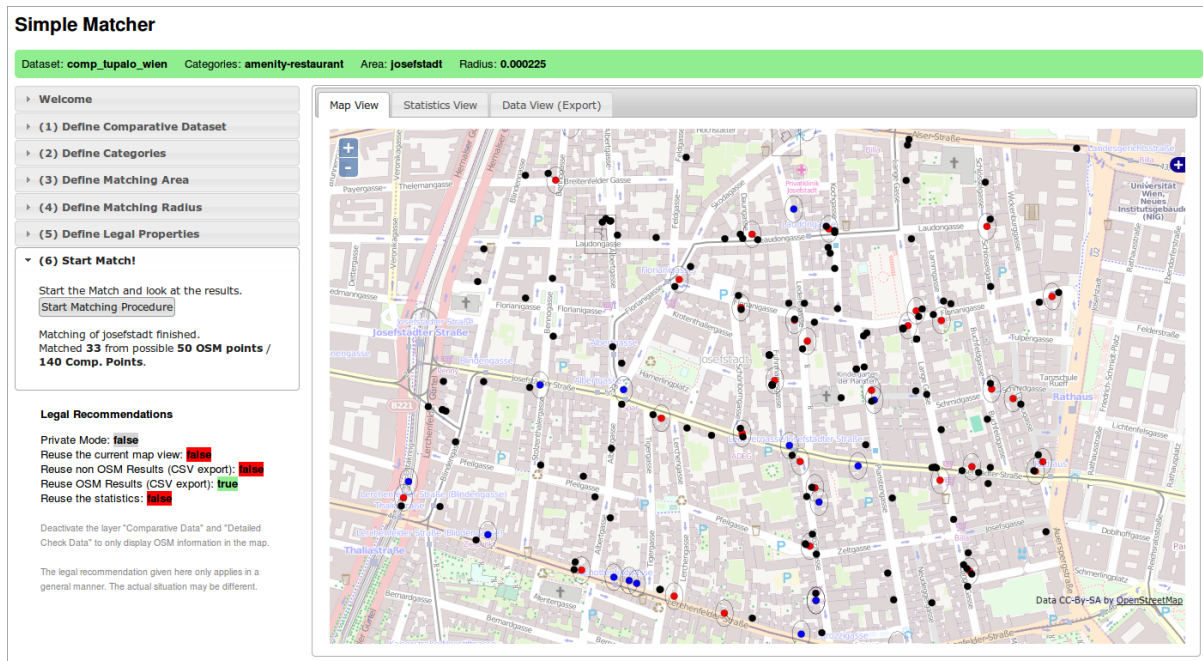


Figure 5.1: The web application after performing a matching procedure

5.3 Operation of the Web Application

The main file of the web application is *index.html* which describes the layout and loads all necessary stylesheets and JavaScript libraries. To create an appealing user interface the jQuery UI library is used [jQuery Foundation, 2012]. For displaying the map OpenLayers [OpenLayers Contributors, 2012] is integrated. Diagrams are created with the Sparkline library [Watts, 2012].

After loading the web application, the map view is initiated and functions to generate user selections are called. Depending on their purpose these functions are stored in the files *logic.js*, *map.js*, *ui.js* and the directory *mod_database*. The generated selections are described in section *User Selections* (page 54). While the user makes his choice, the legal settings and variables needed by the matching module are set according to the previous selection. Also, an information bar on top of the page is updated and changes its colour to indicate when enough information is available to start the matching process.

Now the matching procedure can be started. When the user presses the “Start Matching Procedure” button, the module *mod_m_simple* is initialized. Data coming from the matching module is visualized in the map and a progress bar is shown.

5.3.1 User Selections

The user has to define certain parameters of the matching procedure. These and their functionality are explained here.

When information about the comparative dataset, the categories and the matching area have been entered, data from the database containing the comparative data is requested and visualized on the map. This gives the user an early feedback but may also slow down the responsiveness of the web application when working with big datasets.

Comparative Datasets

First of all the user has to specify the dataset he wants to compare OSM data to (see [figure 7.17](#) (page 105)). This list actually represents single databases that are stored in the same database system with the prefix `comp_`. When loading the page, this list is populated. The user can upload a new dataset by specifying a file in GeoJSON format. A function in `mod_database` then creates a new database and refreshes the list to make the new choice visible.

The refreshing of this list performs very fast since only meta data (a list of all available databases) is requested. The upload depends on the size of the dataset the user has chosen and is not performed per entry but as a bulk upload for the whole GeoJSON document at once.

Categories

The categories by which the OSM dataset is filtered are spatial views that are stored inside the design document of the web application (see section [GeoCouch](#) (page 17)). To populate this list ([figure 7.18](#) (page 105)), the design document is requested and all available views are extracted and displayed.

When the user hovers over a single category a pop-up window displays the content of the view. It is possible to enter a new category selection by specifying a new view. Since these views operate on the OSM dataset which usually will contain lots of entries, it can take an unexpected amount of time to initialize these the first time.

Matching Areas

The areas defining the region the check is performed on are listed in this window ([figure 7.19](#) (page 105)). These entries are single GeoJSON (polygon) documents inside the same database the web application and OSM database are saved to. They can be identified by the key-value `"type" : "SearchPolygon"`. A special view requests all of these and displays them to the user.

When hovering over a single entry the area that is described by this entry is displayed on the map to give the user an idea about the extent. To add a new area, the user has to enter an array of coordinates defining a new area. These coordinates then are stored in a new GeoJSON document of the type `SearchPolygon`.

Search Radius

In this window ([figure 7.20](#) (page 106)) the search radius can be specified in degrees.

Legal Properties

Here (*figure 7.21* (page 106)), the user can specify if he wants to operate in “private” mode and whether the comparative dataset is compatible to the ODbL.

Depending on these choices and on the selection of layers displayed in the map window the legal recommendations (*figure 7.22* (page 106)) are changed by calling functions from the legal module.

5.3.2 Matching

When the user has entered all necessary information, the matching process can be initiated. During this procedure data received from the matching module is visualised on the map and a progress bar shows the progress of the procedure. This happens by the two functions `gjsonToMap()` which is called by the matching module every time a GeoJSON object should be visualised and `updateStatus()` which receives status updates on the matching procedure.

Upon finishing the check, the function `checkFinished()` is called. With this function, the web application creates a new database and stores the matching results in it. When the legal module says that it is not allowed to distribute the comparative data it is not made available to the user.

5.3.3 Matching Results

After completing the matching procedure, three types of results are available:

Map View

On the map (see *figure 5.1* (page 53)) all points from the OSM dataset and the comparative dataset can be visualised. Red points symbolise OSM data that has been matched, blue ones OSM data that has no partner and black points the comparative dataset. Buildings are drawn as outline in any case. Additionally, certain information about the matching procedure like the search radius can be shown. Depending on what the user decides to display the legal recommendations may change.

Statistics View

The statistics view (*figure 7.23* (page 106)) displays some examples on how to further process the results.

At this moment, one can see two pie diagrams giving information about the hit ratio of the OSM and comparative dataset and one bar diagram showing one bar for every OSM user. The height of the bar is calculated from either all points entered by this user that were matched (red, up) or missed (blue, down). It is not possible to generate further diagrams of this kind for any comparative dataset, because a consistency in attributes can not be expected from this source.

Export View

On the export tab, a list of possible exports is shown. These are a CSV file of all missed OSM points, of all matched OSM points or of all matched points from the comparative dataset. Depending on the legal situation the dataset containing the comparative points may be empty. If so, the link is coloured red.

The CSV files contain the coordinates of the points, their `id` within the database, and an entry describing their category (“hit”, “miss”, “hitc”). When the dataset contains matches, there are additionally the `id` of

their matching partner and the distance to it. By providing this information, it is possible to connect the matching pairs afterwards if necessary and also request all information about them from the database by their id.

The CSV files are created with the help of a `list` function inside the CouchDb (see [Anderson et al. \[2010\]](#) in chapter 9 for an explanation of list-functions). By this it is not necessary to programmatically modify the results, they are served directly from the database.

5.4 Testing the System

The tasks described in *Tasks to be Fulfilled* (page 51) have to be performed successfully to verify the hypothesis.

5.4.1 Creation of the Web Application

As can be seen by the sections above, the creation of this web application was successful. It was tested with various browsers. It is operational with Firefox, Opera, Chromium, Epiphany and Safari. The web application itself can be displayed in Internet Explorer, but no match can be performed since the browser of Microsoft does not yet support web workers which was known beforehand (see section *Web Workers* (page 19)). One could emulate the necessary technique by a module simulating a web worker (e.g. [GoogleCode \[2009\]](#)) but this would mean the loss of the advantages provided by it (parallel execution of code) and the browser would crash when performing long matching procedures.

The system is functional as it is when it is stored together with OSM data in one single database. For the operation of the web application additional databases are created automatically, but the application itself does not need more to be reachable or to replicate itself.

Since the web application works with all systems it was expected to work with, both modules could be integrated and a sole database is enough for it to exist. This task was successful.

Task One ✓

5.4.2 Replication of the Web Application

Aside from the computer and database the web application was developed on, it was replicated to another database on the same system which worked without any problems. The same was true for a replication to a CouchDB database inside a VirtualMachine on the same computer. The replication to a completely different system also caused no problems.

Since the GeoCouch extension is not operable on Windows systems yet, it could not be tested whether a CouchDB server running on a Microsoft operating system is capable of serving the web application. What could be determined is that functions that do not rely on the GeoCouch extension are working properly. Thus, one can assume that when this extension becomes available for this operating system the web application will also run on Windows systems.

The system can be replicated, so this task also was successful.

Task Two ✓

5.4.3 Big Amount of Data (National Extent)

There are two datasets involved which may vary in size: the OSM database and the comparative dataset. While it is easy to get OSM data on big extents, it is more difficult with the comparative dataset. The operations performed in this section were executed on a computer running Ubuntu Linux 12.04 with 8GB of memory and a dual core 3.0 GHz CPU.

To test the web application with a big amount of data, the **OSM data extract** of Austria was used. The requirement is to use data of “national extent” which is an ambiguous condition since different nations vary in size. Furthermore, the extent of the area used is only a weak indication of the amount of data that needs to be processed. A small country with a high density of OSM information can be equally challenging as a big country with only a few cities mapped. *Figure 5.2* gives an overview of continents and countries and their density of information. It is important to point out that the size of a dataset not directly represents the number of geometrical entries but also includes attributive information, but it should give a general idea.



Figure 5.2: Extents and sizes

(data taken and generalized from [Geofabrik GmbH \[2012\]](#) and OpenStreetMap, data size is the value of the compressed bz2-file, the ratio value does not take the density of buildings into account, country shapes are in an area accurate azimuthal projection)

One can see that although e.g. Russia spans an area of about 150 times the size of Austria, it only contains 2 to 3 times the amount of data in its OSM extract. Data about France holds ten times more information than the one about Austria. By using the OSM extract of Austria, which is a well mapped country, a fair amount of data is used. Experiences made with this dataset can be related to other countries, since the maximum amount of data available for one country is only up to the factor of 15 of the one of Austria (in case of USA). When removing parts from the extract that are not used by any analysis before importing it into the system, the data size can be reduced drastically. In case of Austria, the data volume can be reduced to a 10th of the original one by removing highways and nodes with no attributes at all.

The compressed extract for Austria is about 364MB in size. After importing it into the CouchDb it contained over 24.000.000 individual documents. The generation of this database required multiple hours but it completed without complications. When actually using the web application with this database for the first time, it is unavoidable that search indices have to be calculated which proves to be the real bottleneck. When the user defines a new search category, he is in fact creating a new view. The database server triggers an index generation when this view is accessed for the first time (which happens already

before starting the actual matching procedure). The generation of a new search index for a new view takes about 20 hours with the OSM dataset of Austria. When the index has already been generated, operating on this large dataset proves to be fast. Aside from the initial index generation the main load on this dataset comes from requesting all points the matching algorithm is iterating over prior to actually starting the matching procedure.

To get a big comparative dataset, the author did not choose to use a dataset with a great extent but with a high density of points. The datasets from Tupalo used in chapter 4 are not used here, because these do not provide a high enough numbers of points and their concentration on urban regions. A **dataset from the Open Data Government initiative of the city of Vienna** [Wien, 2012] was used, namely the cadastre of trees. At this time, it consists of about 120.000 individual entries. This represents a big number of possible comparative points. When one would use all existing branches of e.g. the fast-food restaurant McDonalds worldwide, the amount of entries would sum up to only 33500 points [McDonald's, 2012]. There are three times when the comparative dataset is accessed:

- The first one is when importing it into a new database (see section *Comparative Datasets* (page 54)). The web application can cope with the given amount of data, but when using even bigger datasets, one might think about implementing a new method to create the comparative database to avoid a lag in the user interface during uploading of the data.
- The second time is when the actual comparative data is displayed on the map. This is done by a spatial query that is generated automatically when the database is created. The index generation on this dataset only takes minutes and happens only once. When too much data is rendered to the map, it becomes less responsive. This can be circumvented by deactivating the display of the layer “Comparative Dataset” in the map window.
- The third access happens during the matching procedure. Since the matching module makes use of the same spatial view that was already used when displaying the dataset on the map, the spatial index has already been calculated and there is no lag. The single requests during the matching procedure on this database performed well.

The system has no problems operating on big datasets.

Task Three ✓

5.4.4 Bottlenecks

To better identify problems which could arise using this system, possible bottlenecks are identified. Data used to perform the match is stored in different ways inside the database as is visualised by *figure 5.3*.

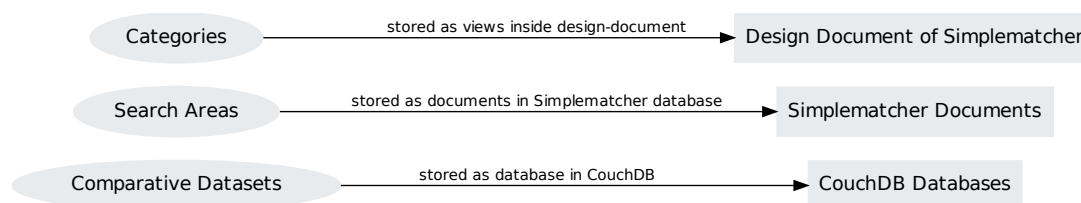


Figure 5.3: Location of data

When starting the web application, various requests on the database are performed and data is loaded into JavaScript variables. As observed before, the most time consuming operation is when the index of a new generated view has to be calculated for the first time. *Figure 5.4* (page 59) gives an overview of different requests to the database.

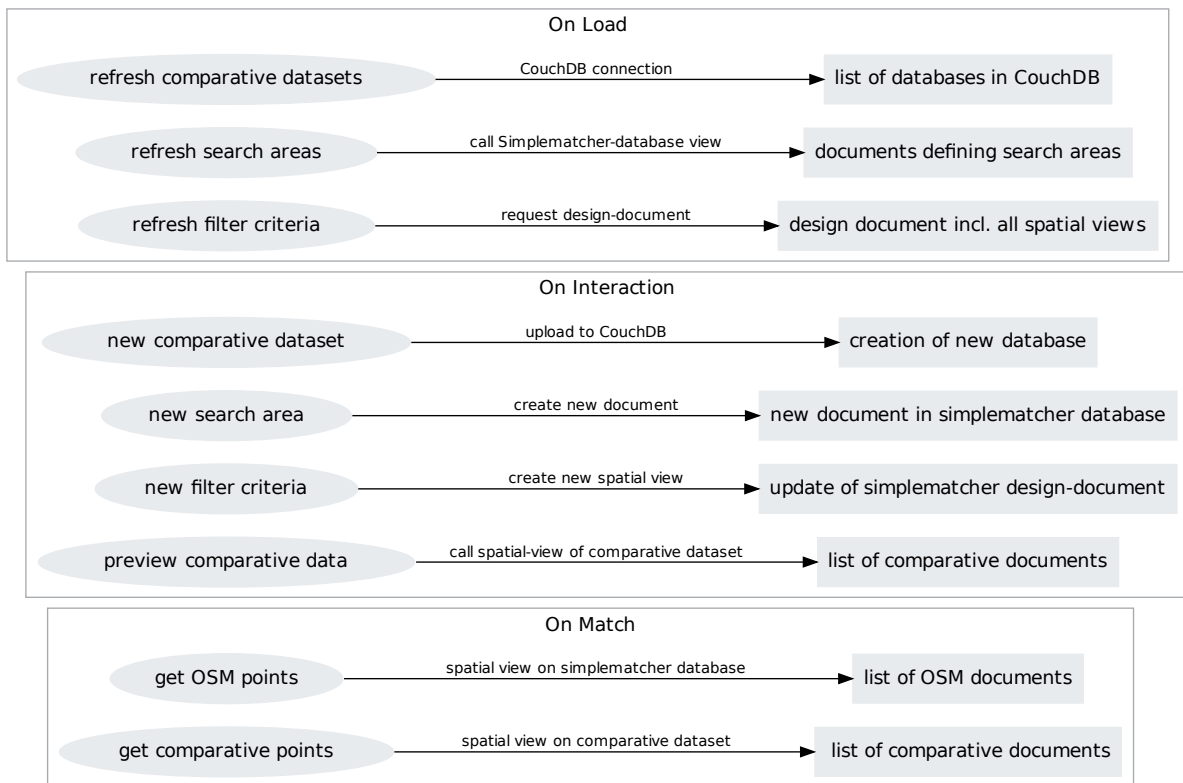


Figure 5.4: Requests to CouchDB

(ellipse) actions of the web application, (line) action performed on database, (rectangles) resulting data

An index may have to be calculated when refreshing search areas, requesting data from the comparative dataset or requesting OSM points. These situations and their occurrences are clarified:

refreshing search areas: This operation is performed each time the web application is loaded or a new search area is defined. When the index for this request has been calculated it performs without noticeable delay.

requesting data from comparative dataset: A spatial view is called during selecting the comparative dataset as well as when starting the matching procedure. Since it is the same spatial view which is called both times the only delay happens when selecting a newly uploaded comparative dataset for the first time.

requesting OSM points: The OSM points are requested by the spatial view the user has defined when starting the matching procedure. The index for this view has to be rebuilt when the user creates a new category to filter OSM points.

Looking at these cases the most troublesome ones are when **requesting data from a newly uploaded comparative dataset or by a newly created spatial view of the OSM dataset**. The selection of the region does not greatly affect the overall responsiveness of the web application.

5.5 Result of this Chapter

All of the tasks described in *Tasks to be Fulfilled* (page 51) could be performed. The system has been programmed, replicated and tested with big datasets. The hypothesis of this chapter is verified and certain bottlenecks when using a system architecture like this could be identified. These mainly concern

the index generation when adding a new selection of categories of the Open Street Map dataset. Since this web application is a reference implementation there is room left for optimisation. Even when using this canonical form of interaction with the database and the matching algorithm, it performs sufficiently well when not defining a new set of categories. For doing a casual analysis, the need for hard drive space of a CouchDB could be a problem. For the dataset of Austria including generated views, the required disk space is about 9 GB. By filtering out elements before creating the OSM database, its size can be reduced (see section *Preparing the Database* (page 52)).

The complete web application and OSM data is saved in one single database which can be replicated to any other CouchDB without problems. To actually make the web application work, the CouchDB installation has to include the GeoCouch extension. But this does not concern the simple storing of the web application which works in any case.

By generating exemplary statistics (see section *Statistics View* (page 55)), a possible way to make further use of the matched data is shown. The data made available after the matching procedure is not reduced in information which means that all Open Street Map attributes that were associated with a single entry are also available in the result set.

CONCLUSION

Content of this Chapter

The analysis program itself as well as its results are reviewed bearing the initial research hypotheses in mind. Also, possible limitations and optimisations are indicated.

6.1 Results

The results of the analysis performed in each chapter are listed according to their corresponding research hypothesis.

(Chapter 3) **By paying attention to legal guidelines it is possible to declare the legal properties of the results programmatically.**

All licenses involved have been discussed and their influence on the matching procedure has been evaluated. A module has been programmed that provides information about what the user is allowed to do with the results. This information depends on previously defined legal circumstances of the involved datasets. This module serves as a prototype that gives estimated results. Its function shows that it is possible to construct a framework which can give even more precise legal advice.

By integrating this legal module into the final web application, this hypothesis can be confirmed.

(Chapter 4) **By using a spatially enabled document-based database and a well defined input dataset, a simplified geographical comparison algorithm can produce acceptable results.**

A simplified matching algorithm has been defined, constructed and evaluated. The dataset used for comparison was a set of restaurants in two major cities in Austria provided by Tupalo [Tupalo Internet-services GmbH, 2012]. The results were acceptable (see *Discussion of the Criteria* (page 47)) for three of the four tested cases (two districts in Vienna and two districts in Graz). One test did not reach the expected accuracy possible because of the suspected excessively high density of points. A more complex matching algorithm (see section *Further Optimisation* (page 43)) may produce better results.

While the simplified matching algorithm may be of alternate use, this hypothesis has to be rejected.

(Chapter 5) **The complete system can be contained within one independent, single database and perform a large-scale analysis in an efficient manner.**

A canonical implementation of a web application making use of the legal module and the matching module has been made. This application as well as the needed OSM dataset are stored in one single CouchDB. Exemplary analysis with two extensive datasets have been performed and the functionality

of the web application could be proven (the term “big/national extent” is explained in section *Big Amount of Data (National Extent)* (page 57)). Further optimisation of this program is possible.

A system as described in the hypothesis was built successfully and can perform up to a national extent. Therefore this hypothesis can be confirmed.

“It is possible to construct a system to perform a simplified comparative analysis of specific OpenStreetMap – POIs with previously selected datasets by using a document-based database.”

By evaluating the matching algorithm, it has been shown that a system such as this is capable of performing an analysis as stated by the hypothesis. By having additionally confirmed two of the above three hypotheses (creation of a web application in one single database, legal module), the main hypothesis of the overall thesis can be verified. The server requires a single software (CouchDB), while on the side of the client a recent web browser with support of HTML5 suffices to use the matching program.

6.2 Optimisation and Further Research

When taking a deeper look at the legal context of the OpenStreetMap project, a more general module giving legal recommendations could be developed. This module may even cover more use-cases than just OpenStreetMap. This could concern the field of “legal ontologies”.

There are weaknesses associated with using a simplified matching algorithm. It can only perform as well as has been shown by this thesis. Multiple optimisations have been laid out which have different effects on the kind of matching errors they are able to avoid. These optimisations require further testing and evaluation. Also, the simplified matching algorithm may perform differently when used in more rural regions. Since the search radius is the only parameter of the matching process, the user can choose freely without dependency on any restrictions of category or place, a connection between the measured average minimum distance of the points of the comparative dataset and the optimal search radius could also be explored.

While it has been shown that the construction of a web application like the one proposed contained within a CouchDB database is already possible, further development of this database system will facilitate the integration of such web applications. Some performance glitches have been identified. These exist mainly because the user is able to change many variables (like defining a new view) when performing a new match. When utilizing less flexibility, a CouchDB database may perform better. Based on the findings of this thesis, a classification of typical web mapping applications and their requirements could be done to find out for which cases it is advisable to use a CouchDB as program and database storage system. This database system is further developed and optimised, but there are some principal limitations that will not change in foreseeable future (like the generation of view indices).

There is potential in using this kind of system architecture. A simplified matching algorithm may produce usable results when optimized. The legal circumstances can be brought to attention when evaluating the input datasets.

APPENDIX

Contents

7.1	Legal Texts	63
7.1.1	Contributor Terms 1.2.4	63
7.1.2	Creative Commons Share Alike (CC-BY-SA) 2.0	65
7.1.3	Open Database Licence (ODbL) 1.0	68
7.1.4	Database Content Licence (OdCL)	78
7.2	Source Code	79
7.2.1	Source Code of the Legal Module	79
7.2.2	Source Code of the Matching Module	80
7.3	Comparison Maps	88
7.4	Pictures of the Web Application	105

7.1 Legal Texts

7.1.1 Contributor Terms 1.2.4

source: http://www.osmfoundation.org/wiki/License/Contributor_Terms

Thank you for your interest in contributing data and/or any other content (collectively, "Contents") to the geo-database of the OpenStreetMap project (the "Project"). This contributor agreement (the "Agreement") is made between you ("You") and The OpenStreetMap Foundation ("OSMF") and clarifies the intellectual property rights in any Contents that You choose to submit to the Project in this user account. Please read the following terms and conditions carefully and click either the "Accept" or "Decline" button at the bottom to continue.

1. We respect the intellectual property rights of others and we need to be able to respond to any objections by intellectual property owners. This means that:

(a) Your contribution of data should not infringe the intellectual property rights of anyone else. If you contribute Contents, You are indicating that, as far as You know, You have the right to authorize OSMF to use and distribute those Contents under our current licence terms. If You do not have that right, You risk having Your contribution deleted (see below).

(b) Please note that OSMF does not have to include Contents You contribute in

the Project, and may remove Your contributions from the Project at any time. For example, if we suspect that any contributed data is incompatible, (in the sense that we could not continue to lawfully distribute it), with whichever licence or licences we are then using (see sections 3 and 4), then we may delete that data.

Rights Granted

2. Subject to Section 3 and 4 below, You hereby grant to OSMF a worldwide, royalty-free, non-exclusive, perpetual, irrevocable licence to do any act that is restricted by copyright, database right or any related right over anything within the Contents, whether in the original medium or any other. These rights explicitly include commercial use, and do not exclude any field of endeavour. These rights include, without limitation, the right to sub-license the work through multiple tiers of sub-licensees and to sue for any copyright violation directly connected with OSMF's rights under these terms. To the extent allowable under applicable local laws and copyright conventions, You also waive and/or agree not to assert against OSMF or its licensees any moral rights that You may have in the Contents.

3. OSMF agrees that it may only use or sub-license Your Contents as part of a database and only under the terms of one or more of the following licences: ODbL 1.0 for the database and DbCL 1.0 for the individual contents of the database; CC-BY-SA 2.0; or such other free and open licence (for example, <http://www.opendefinition.org/okd/>) as may from time to time be chosen by a vote of the OSMF membership and approved by at least a 2/3 majority vote of active contributors.

An "active contributor" is defined as:

a natural person (whether using a single or multiple accounts) who has edited the Project in any three calendar months from the last 12 months (i.e. there is a demonstrated interest over time); and has maintained a valid email address in their registration profile and responds to a request to vote within 3 weeks.

4. At Your or the copyright owner's option, OSMF agrees to attribute You or the copyright owner. A mechanism will be provided, currently a web page <http://wiki.openstreetmap.org/wiki/Attribution>.

5. Except as set forth herein, You reserve all right, title, and interest in and to Your Contents.

Limitation of Liability

6. To the extent permitted by applicable law, You provide the Contents "as is" without warranty of any kind, either express or implied, including without limitation any warranties or conditions of merchantability, fitness for a particular purpose, or otherwise.

7. Subject to any liability that may not be excluded or limited by law, neither You nor OSMF shall be liable for any special, indirect, incidental, consequential, punitive, or exemplary damages under this Agreement, however caused and under any theory of liability. This exclusion applies even if either party has been advised of the possibility of such damages.

Miscellaneous

8. This Agreement shall be governed by English law without regard to principles of conflict of law. You agree that the United Nations Convention on Contracts

for the International Sale of Goods (1980) is hereby excluded in its entirety from application to this Agreement. In the event of invalidity of any provision of this Agreement, the parties agree that such invalidity shall not affect the validity of the remaining portions of this Agreement. This is the entire agreement between You and OSMF which supersedes any prior agreement, whether written, oral or other, relating to the subject matter of this agreement.

7.1.2 Creative Commons Share Alike (CC-BY-SA) 2.0

source: <http://creativecommons.org/licenses/by-sa/2.0/legalcode>

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

"Collective Work" means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.

"Derivative Work" means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.

"Licensor" means the individual or entity that offers the Work under the terms of this License.

"Original Author" means the individual or entity who created the Work.

"Work" means the copyrightable work of authorship offered under the terms of this License.

"You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

"License Elements" means the following high-level license attributes as selected by Licensor and indicated in the title of this License: Attribution,

ShareAlike.

2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;
to create and reproduce Derivative Works;
to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;
to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission Derivative Works.
For the avoidance of doubt, where the work is a musical composition:

Performance Royalties Under Blanket Licenses. Licensor waives the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work.

Mechanical Rights and Statutory Royalties. Licensor waives the exclusive right to collect, whether individually or via a music rights society or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions).

Webcasting Rights and Statutory Royalties. For the avoidance of doubt, where the Work is a sound recording, Licensor waives the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions). The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. All rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that

control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any reference to such Licensor or the Original Author, as requested. If You create a Derivative Work, upon notice from any Licensor You must, to the extent practicable, remove from the Derivative Work any reference to such Licensor or the Original Author, as requested.

You may distribute, publicly display, publicly perform, or publicly digitally perform a Derivative Work only under the terms of this License, a later version of this License with the same License Elements as this License, or a Creative Commons iCommons license that contains the same License Elements as this License (e.g. Attribution-ShareAlike 2.0 Japan). You must include a copy of, or the Uniform Resource Identifier for, this License or other license specified in the previous sentence with every copy or phonorecord of each Derivative Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Derivative Works that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder, and You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Derivative Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Derivative Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Derivative Work itself to be made subject to the terms of this License.

If you distribute, publicly display, publicly perform, or publicly digitally perform the Work or any Derivative Works or Collective Works, You must keep intact all copyright notices for the Work and give the Original Author credit reasonable to the medium or means You are utilizing by conveying the name (or pseudonym if applicable) of the Original Author if supplied; the title of the Work if supplied; to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and in the case of a Derivative Work, a credit identifying the use of the Work in the Derivative Work (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Derivative Work or Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE MATERIALS, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW,

IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

7.1.3 Open Database Licence (ODbL) 1.0

source: <http://opendatacommons.org/licenses/odbl/1.0/>

```
## ODC Open Database License (ODbL)
```

```
### Preamble
```

The Open Database License (ODbL) is a license agreement intended to allow users to freely share, modify, and use this Database while maintaining this same freedom for others. Many databases are covered by

copyright, and therefore this document licenses these rights. Some jurisdictions, mainly in the European Union, have specific rights that cover databases, and so the ODbL addresses these rights, too. Finally, the ODbL is also an agreement in contract for users of this Database to act in certain ways in return for accessing this Database.

Databases can contain a wide variety of types of content (images, audiovisual material, and sounds all in the same database, for example), and so the ODbL only governs the rights over the Database, and not the contents of the Database individually. Licensors should use the ODbL together with another license for the contents, if the contents have a single set of rights that uniformly covers all of the contents. If the contents have multiple sets of different rights, Licensors should describe what rights govern what contents together in the individual record or in some other way that clarifies what rights apply.

Sometimes the contents of a database, or the database itself, can be covered by other rights not addressed here (such as private contracts, trade mark over the name, or privacy rights / data protection rights over information in the contents), and so you are advised that you may have to consult other documents or clear other rights before doing activities not covered by this License.

The Licensor (as defined below)

and

You (as defined below)

agree as follows:

1.0 Definitions of Capitalised Words

"Collective Database" - Means this Database in unmodified form as part of a collection of independent databases in themselves that together are assembled into a collective whole. A work that constitutes a Collective Database will not be considered a Derivative Database.

"Convey" - As a verb, means Using the Database, a Derivative Database, or the Database as part of a Collective Database in any way that enables a Person to make or receive copies of the Database or a Derivative Database. Conveying does not include interaction with a user through a computer network, or creating and Using a Produced Work, where no transfer of a copy of the Database or a Derivative Database occurs.

"Contents" - The contents of this Database, which includes the information, independent works, or other material collected into the Database. For example, the contents of the Database could be factual data or works such as images, audiovisual material, text, or sounds.

"Database" - A collection of material (the Contents) arranged in a systematic or methodical way and individually accessible by electronic or other means offered under the terms of this License.

"Database Directive" - Means Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection

of databases, as amended or succeeded.

"Database Right" - Means rights resulting from the Chapter III ("sui generis") rights in the Database Directive (as amended and as transposed by member states), which includes the Extraction and Re-utilisation of the whole or a Substantial part of the Contents, as well as any similar rights available in the relevant jurisdiction under Section 10.4.

"Derivative Database" - Means a database based upon the Database, and includes any translation, adaptation, arrangement, modification, or any other alteration of the Database or of a Substantial part of the Contents. This includes, but is not limited to, Extracting or Re-utilising the whole or a Substantial part of the Contents in a new Database.

"Extraction" - Means the permanent or temporary transfer of all or a Substantial part of the Contents to another medium by any means or in any form.

"License" - Means this license agreement and is both a license of rights such as copyright and Database Rights and an agreement in contract.

"Licensor" - Means the Person that offers the Database under the terms of this License.

"Person" - Means a natural or legal person or a body of persons corporate or incorporate.

"Produced Work" - a work (such as an image, audiovisual material, text, or sounds) resulting from using the whole or a Substantial part of the Contents (via a search or other query) from this Database, a Derivative Database, or this Database as part of a Collective Database.

"Publicly" - means to Persons other than You or under Your control by either more than 50% ownership or by the power to direct their activities (such as contracting with an independent consultant).

"Re-utilisation" - means any form of making available to the public all or a Substantial part of the Contents by the distribution of copies, by renting, by online or other forms of transmission.

"Substantial" - Means substantial in terms of quantity or quality or a combination of both. The repeated and systematic Extraction or Re-utilisation of insubstantial parts of the Contents may amount to the Extraction or Re-utilisation of a Substantial part of the Contents.

"Use" - As a verb, means doing any act that is restricted by copyright or Database Rights whether in the original medium or any other; and includes without limitation distributing, copying, publicly performing, publicly displaying, and preparing derivative works of the Database, as well as modifying the Database as may be technically necessary to use it in a different mode or format.

"You" - Means a Person exercising rights under this License who has not previously violated the terms of this License with respect to the Database, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

Words in the singular include the plural and vice versa.

2.0 What this License covers

2.1. Legal effect of this document. This License is:

- a. A license of applicable copyright and neighbouring rights;
- b. A license of the Database Right; and
- c. An agreement in contract between You and the Licensor.

2.2 Legal rights covered. This License covers the legal rights in the Database, including:

- a. Copyright. Any copyright or neighbouring rights in the Database. The copyright licensed includes any individual elements of the Database, but does not cover the copyright over the Contents independent of this Database. See Section 2.4 for details. Copyright law varies between jurisdictions, but is likely to cover: the Database model or schema, which is the structure, arrangement, and organisation of the Database, and can also include the Database tables and table indexes; the data entry and output sheets; and the Field names of Contents stored in the Database;
- b. Database Rights. Database Rights only extend to the Extraction and Re-utilisation of the whole or a Substantial part of the Contents. Database Rights can apply even when there is no copyright over the Database. Database Rights can also apply when the Contents are removed from the Database and are selected and arranged in a way that would not infringe any applicable copyright; and
- c. Contract. This is an agreement between You and the Licensor for access to the Database. In return you agree to certain conditions of use on this access as outlined in this License.

2.3 Rights not covered.

- a. This License does not apply to computer programs used in the making or operation of the Database;
- b. This License does not cover any patents over the Contents or the Database; and
- c. This License does not cover any trademarks associated with the Database.

2.4 Relationship to Contents in the Database. The individual items of the Contents contained in this Database may be covered by other rights, including copyright, patent, data protection, privacy, or personality rights, and this License does not cover any rights (other than Database Rights or in contract) in individual Contents contained in the Database. For example, if used on a Database of images (the Contents), this License would not apply to copyright over individual images, which could have their own separate licenses, or one single license covering all of the rights over the images.

3.0 Rights granted

3.1 Subject to the terms and conditions of this License, the Licensor grants to You a worldwide, royalty-free, non-exclusive, terminable (but only under Section 9) license to Use the Database for the duration of any applicable copyright and Database Rights. These rights explicitly include commercial use, and do not exclude any field of endeavour. To the extent possible in the relevant jurisdiction, these rights may be exercised in all media and formats whether now known or created in the future.

The rights granted cover, for example:

- a. Extraction and Re-utilisation of the whole or a Substantial part of the Contents;
- b. Creation of Derivative Databases;
- c. Creation of Collective Databases;
- d. Creation of temporary or permanent reproductions by any means and in any form, in whole or in part, including of any Derivative Databases or as a part of Collective Databases; and
- e. Distribution, communication, display, lending, making available, or performance to the public by any means and in any form, in whole or in part, including of any Derivative Database or as a part of Collective Databases.

3.2 Compulsory license schemes. For the avoidance of doubt:

- a. Non-waivable compulsory license schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;
- b. Waivable compulsory license schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,
- c. Voluntary license schemes. The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

3.3 The right to release the Database under different terms, or to stop distributing or making available the Database, is reserved. Note that this Database may be multiple-licensed, and so You may have the choice of using alternative licenses for this Database. Subject to Section 10.4, all other rights not expressly granted by Licensor are reserved.

4.0 Conditions of Use

4.1 The rights granted in Section 3 above are expressly made subject to Your complying with the following conditions of use. These are important conditions of this License, and if You fail to follow them, You will be in material breach of its terms.

4.2 Notices. If You Publicly Convey this Database, any Derivative Database, or the Database as part of a Collective Database, then You must:

- a. Do so only under the terms of this License or another license permitted under Section 4.4;
- b. Include a copy of this License (or, as applicable, a license permitted under Section 4.4) or its Uniform Resource Identifier (URI) with the Database or Derivative Database, including both in the Database or Derivative Database and in any relevant documentation; and
- c. Keep intact any copyright or Database Right notices and notices that refer to this License.
- d. If it is not possible to put the required notices in a particular file due to its structure, then You must include the notices in a location (such as a relevant directory) where users would be likely to look for it.

4.3 Notice for using output (Contents). Creating and Using a Produced Work does not require the notice in Section 4.2. However, if you Publicly Use a Produced Work, You must include a notice associated with the Produced Work reasonably calculated to make any Person that uses, views, accesses, interacts with, or is otherwise exposed to the Produced Work aware that Content was obtained from the Database, Derivative Database, or the Database as part of a Collective Database, and that it is available under this License.

- a. Example notice. The following text will satisfy notice under Section 4.3:

Contains information from DATABASE NAME, which is made available here under the Open Database License (ODbL).

DATABASE NAME should be replaced with the name of the Database and a hyperlink to the URI of the Database. "Open Database License" should contain a hyperlink to the URI of the text of this License. If hyperlinks are not possible, You should include the plain text of the required URI's with the above notice.

4.4 Share alike.

- a. Any Derivative Database that You Publicly Use must be only under the terms of:
 - i. This License;
 - ii. A later version of this License similar in spirit to this License; or
 - iii. A compatible license.

If You license the Derivative Database under one of the licenses mentioned in (iii), You must comply with the terms of that license.

b. For the avoidance of doubt, Extraction or Re-utilisation of the whole or a Substantial part of the Contents into a new database is a Derivative Database and must comply with Section 4.4.

c. Derivative Databases and Produced Works. A Derivative Database is Publicly Used and so must comply with Section 4.4. if a Produced Work created from the Derivative Database is Publicly Used.

d. Share Alike and additional Contents. For the avoidance of doubt, You must not add Contents to Derivative Databases under Section 4.4 a that are incompatible with the rights granted under this License.

e. Compatible licenses. Licensors may authorise a proxy to determine compatible licenses under Section 4.4 a iii. If they do so, the authorised proxy's public statement of acceptance of a compatible license grants You permission to use the compatible license.

4.5 Limits of Share Alike. The requirements of Section 4.4 do not apply in the following:

a. For the avoidance of doubt, You are not required to license Collective Databases under this License if You incorporate this Database or a Derivative Database in the collection, but this License still applies to this Database or a Derivative Database as a part of the Collective Database;

b. Using this Database, a Derivative Database, or this Database as part of a Collective Database to create a Produced Work does not create a Derivative Database for purposes of Section 4.4; and

c. Use of a Derivative Database internally within an organisation is not to the public and therefore does not fall under the requirements of Section 4.4.

4.6 Access to Derivative Databases. If You Publicly Use a Derivative Database or a Produced Work from a Derivative Database, You must also offer to recipients of the Derivative Database or Produced Work a copy in a machine readable form of:

a. The entire Derivative Database; or

b. A file containing all of the alterations made to the Database or the method of making the alterations to the Database (such as an algorithm), including any additional Contents, that make up all the differences between the Database and the Derivative Database.

The Derivative Database (under a.) or alteration file (under b.) must be available at no more than a reasonable production cost for physical distributions and free of charge if distributed over the internet.

4.7 Technological measures and additional terms

a. This License does not allow You to impose (except subject to

Section 4.7 b.) any terms or any technological measures on the Database, a Derivative Database, or the whole or a Substantial part of the Contents that alter or restrict the terms of this License, or any rights granted under it, or have the effect or intent of restricting the ability of any person to exercise those rights.

b. Parallel distribution. You may impose terms or technological measures on the Database, a Derivative Database, or the whole or a Substantial part of the Contents (a "Restricted Database") in contravention of Section 4.74 a. only if You also make a copy of the Database or a Derivative Database available to the recipient of the Restricted Database:

- i. That is available without additional fee;
- ii. That is available in a medium that does not alter or restrict the terms of this License, or any rights granted under it, or have the effect or intent of restricting the ability of any person to exercise those rights (an "Unrestricted Database"); and
- iii. The Unrestricted Database is at least as accessible to the recipient as a practical matter as the Restricted Database.

c. For the avoidance of doubt, You may place this Database or a Derivative Database in an authenticated environment, behind a password, or within a similar access control scheme provided that You do not alter or restrict the terms of this License or any rights granted under it or have the effect or intent of restricting the ability of any person to exercise those rights.

4.8 Licensing of others. You may not sublicense the Database. Each time You communicate the Database, the whole or Substantial part of the Contents, or any Derivative Database to anyone else in any way, the Licensor offers to the recipient a license to the Database on the same terms and conditions as this License. You are not responsible for enforcing compliance by third parties with this License, but You may enforce any rights that You have over a Derivative Database. You are solely responsible for any modifications of a Derivative Database made by You or another Person at Your direction. You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License.

5.0 Moral rights

5.1 Moral rights. This section covers moral rights, including any rights to be identified as the author of the Database or to object to treatment that would otherwise prejudice the author's honour and reputation, or any other derogatory treatment:

- a. For jurisdictions allowing waiver of moral rights, Licensor waives all moral rights that Licensor may have in the Database to the fullest extent possible by the law of the relevant jurisdiction under Section 10.4;
- b. If waiver of moral rights under Section 5.1 a in the relevant jurisdiction is not possible, Licensor agrees not to assert any moral rights over the Database and waives all claims in moral rights to the

fullest extent possible by the law of the relevant jurisdiction under Section 10.4; and

c. For jurisdictions not allowing waiver or an agreement not to assert moral rights under Section 5.1 a and b, the author may retain their moral rights over certain aspects of the Database.

Please note that some jurisdictions do not allow for the waiver of moral rights, and so moral rights may still subsist over the Database in some jurisdictions.

6.0 Fair dealing, Database exceptions, and other rights not affected

6.1 This License does not affect any rights that You or anyone else may independently have under any applicable law to make any use of this Database, including without limitation:

a. Exceptions to the Database Right including: Extraction of Contents from non-electronic Databases for private purposes, Extraction for purposes of illustration for teaching or scientific research, and Extraction or Re-utilisation for public security or an administrative or judicial procedure.

b. Fair dealing, fair use, or any other legally recognised limitation or exception to infringement of copyright or other applicable laws.

6.2 This License does not affect any rights of lawful users to Extract and Re-utilise insubstantial parts of the Contents, evaluated quantitatively or qualitatively, for any purposes whatsoever, including creating a Derivative Database (subject to other rights over the Contents, see Section 2.4). The repeated and systematic Extraction or Re-utilisation of insubstantial parts of the Contents may however amount to the Extraction or Re-utilisation of a Substantial part of the Contents.

7.0 Warranties and Disclaimer

7.1 The Database is licensed by the Licensor "as is" and without any warranty of any kind, either express, implied, or arising by statute, custom, course of dealing, or trade usage. Licensor specifically disclaims any and all implied warranties or conditions of title, non-infringement, accuracy or completeness, the presence or absence of errors, fitness for a particular purpose, merchantability, or otherwise. Some jurisdictions do not allow the exclusion of implied warranties, so this exclusion may not apply to You.

8.0 Limitation of liability

8.1 Subject to any liability that may not be excluded or limited by law, the Licensor is not liable for, and expressly excludes, all liability for loss or damage however and whenever caused to anyone by any use under this License, whether by You or by anyone else, and whether caused by any fault on the part of the Licensor or not. This exclusion of liability includes, but is not limited to, any special, incidental, consequential, punitive, or exemplary damages such as loss of revenue, data, anticipated profits, and lost business. This exclusion applies even if the Licensor has been advised of the possibility of such

damages.

8.2 If liability may not be excluded by law, it is limited to actual and direct financial loss to the extent it is caused by proved negligence on the part of the Licensor.

9.0 Termination of Your rights under this License

9.1 Any breach by You of the terms and conditions of this License automatically terminates this License with immediate effect and without notice to You. For the avoidance of doubt, Persons who have received the Database, the whole or a Substantial part of the Contents, Derivative Databases, or the Database as part of a Collective Database from You under this License will not have their licenses terminated provided their use is in full compliance with this License or a license granted under Section 4.8 of this License. Sections 1, 2, 7, 8, 9 and 10 will survive any termination of this License.

9.2 If You are not in breach of the terms of this License, the Licensor will not terminate Your rights under it.

9.3 Unless terminated under Section 9.1, this License is granted to You for the duration of applicable rights in the Database.

9.4 Reinstatement of rights. If you cease any breach of the terms and conditions of this License, then your full rights under this License will be reinstated:

- a. Provisionally and subject to permanent termination until the 60th day after cessation of breach;
- b. Permanently on the 60th day after cessation of breach unless otherwise reasonably notified by the Licensor; or
- c. Permanently if reasonably notified by the Licensor of the violation, this is the first time You have received notice of violation of this License from the Licensor, and You cure the violation prior to 30 days after your receipt of the notice.

Persons subject to permanent termination of rights are not eligible to be a recipient and receive a license under Section 4.8.

9.5 Notwithstanding the above, Licensor reserves the right to release the Database under different license terms or to stop distributing or making available the Database. Releasing the Database under different license terms or stopping the distribution of the Database will not withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

10.0 General

10.1 If any provision of this License is held to be invalid or unenforceable, that must not affect the validity or enforceability of the remainder of the terms and conditions of this License and each remaining provision of this License shall be valid and enforced to the

fullest extent permitted by law.

10.2 This License is the entire agreement between the parties with respect to the rights granted here over the Database. It replaces any earlier understandings, agreements or representations with respect to the Database.

10.3 If You are in breach of the terms of this License, You will not be entitled to rely on the terms of this License or to complain of any breach by the Licensor.

10.4 Choice of law. This License takes effect in and will be governed by the laws of the relevant jurisdiction in which the License terms are sought to be enforced. If the standard suite of rights granted under applicable copyright law and Database Rights in the relevant jurisdiction includes additional rights not granted under this License, these additional rights are granted in this License in order to meet the terms of this License.

7.1.4 Database Content Licence (OdCL)

source: <http://opendatacommons.org/licenses/dbcl/1.0/>

ODC Database Contents License

The Licensor and You agree as follows:

1.0 Definitions of Capitalised Words

The definitions of the Open Database License (ODbL) 1.0 are incorporated by reference into the Database Contents License.

2.0 Rights granted and Conditions of Use

2.1 Rights granted. The Licensor grants to You a worldwide, royalty-free, non-exclusive, perpetual, irrevocable copyright license to do any act that is restricted by copyright over anything within the Contents, whether in the original medium or any other. These rights explicitly include commercial use, and do not exclude any field of endeavour. These rights include, without limitation, the right to sublicense the work.

2.2 Conditions of Use. You must comply with the ODbL.

2.3 Relationship to Databases and ODbL. This license does not cover any Database Rights, Database copyright, or contract over the Contents as part of the Database. Please see the ODbL covering the Database for more details about Your rights and obligations.

2.4 Non-assertion of copyright over facts. The Licensor takes the position that factual information is not covered by copyright. The DbCL grants you permission for any information having copyright contained in the Contents.

3.0 Warranties, disclaimer, and limitation of liability

3.1 The Contents are licensed by the Licensor "as is" and without any warranty of any kind, either express or implied, whether of title, of accuracy, of the presence of absence of errors, of fitness for purpose, or otherwise. Some jurisdictions do not allow the exclusion of implied warranties, so this exclusion may not apply to You.

3.2 Subject to any liability that may not be excluded or limited by law, the Licensor is not liable for, and expressly excludes, all liability for loss or damage however and whenever caused to anyone by any use under this License, whether by You or by anyone else, and whether caused by any fault on the part of the Licensor or not. This exclusion of liability includes, but is not limited to, any special, incidental, consequential, punitive, or exemplary damages. This exclusion applies even if the Licensor has been advised of the possibility of such damages.

3.3 If liability may not be excluded by law, it is limited to actual and direct financial loss to the extent it is caused by proved negligence on the part of the Licensor.

7.2 Source Code

7.2.1 Source Code of the Legal Module

```
1  var mod_legal = (function() {
2    return{
3      // using the map resulting from the analysis
4      allowMapCopy : function(settings){
5        if(settings.privateMode){ return(true) };
6        if(settings.comparativeInMap){
7          if(settings.comparativeOdblCompatible){
8            return(true);
9          }else{ return(false); };
10       }else{ return(true); };
11     }, //allowMapCopy
12
13     // using the data-results of the analysis
14     allowDataCopy : function(settings){
15       if(settings.privateMode){ return(true)};
16       if(settings.comparativeInData){
17         if(settings.comparativeOdblCompatible){
18           return(true);
19         }else{ return(false); };
20       }else{ return(true); };
21     }, //allowResultDistribution
22
23     // using visual statistics resulting from the analysis
24     allowDiagramCopy : function(settings){
25       if(settings.privateMode){ return(true)};
26       if(settings.comparativeInDiagram){
27         if(settings.comparativeOdblCompatible){
28           return(true);
29         }else{ return(false); };
30       }else{ return(true); };

```

```
31     }, //allowStatCopy
32   }; //return
33 }) ();
```

7.2.2 Source Code of the Matching Module

Source Code of the Main Module

```
1  var mod_m_simple = (function() {
2    var module_status;
3    var module_percent = 0;
4    var worker;
5
6    return{
7      start : function(checkarray, settings) {
8        module_status = 1;
9        //console.log("mod_m_simple.start()");
10       checkarray = input2array(checkarray);
11
12       // start the web-worker
13       worker = new Worker("mod_m_simple/worker.js");
14       var workerconnection = settings.connectionstring;
15       worker.postMessage( {   command : "start",
16                             checksettings : settings,
17                             content : checkarray,
18                             connection : workerconnection
19                           });
20
21       // call the "updateStatus" function provided by the main web-app
22       updateStatus({title : "Matching ...", status : 0});
23
24       // worker callback
25       worker.onmessage = function(message) {
26         if (message.data.command == "info") {
27           console.log("worker says: " + message.data.content);
28         } else if (message.data.command == "status") {
29           module_percent = message.data.content;
30           updateStatus({title : "Matching ...", status : message.data.content});
31         } else if (message.data.command == "finished") {
32           console.log("WORKER HAS FINISHED");
33           var lencheck = message.data.content.hit.length;
34           var lenmiss = message.data.content.miss.length;
35           updateStatus({title : "Matching",
36                       status : "Found <b>" + lencheck +
37                                " matches </b> from possible " +
38                                (lencheck+lenmiss) + ".<br/> "
39                             });
40           checkFinished(message.data.content);
41           worker.terminate();
42           delete worker;
43         } else if (message.data.command == "inspect") {
44           console.log("Worker sends the following object to inspect:");
45           console.log(message.data.content);
46         } else if (message.data.command == "addToMap") {
47           //console.log(message.data.content);
```



```

48     gjsonToMap( message.data.content); //this function is external
49     var drawlayer = message.data.content.layer;
50     // only draw the search-extent when it was actually a search-element
51     if (drawlayer == "hit" || drawlayer == "miss") {
52         drawSearchExtent( message.data.content);
53     };
54 };
55 }; // END worker.onmessage
56 module_status = 0;
57 return(0);
58 }, //start
59
60 status : function() {
61     if (module_status == 1) {
62         return(this.module_percent);
63     };
64 }, //status
65
66 stop : function() {
67     // stop the web-worker
68     if(worker) {
69         worker.terminate();
70         module_status = 0;
71         delete worker;
72         return ("stopped");
73     };
74 } //stop
75 }; //return
76
77 // draws a circle or a box around a specified feature and pushes it to the map
78 function drawSearchExtent( feature) {
79     if (settings.radiussearch) {
80         // build a circle
81         var bboxgjson = buildRadius( feature, settings.searchradius);
82         bboxgjson.layer = "checkdata";
83     } else {
84         // build a box
85         var bboxgjson = buildGjsonBox( feature, settings.searchradius);
86         bboxgjson.layer = "checkdata";
87     };
88     gjsonToMap( bboxgjson); //this function must be present externally
89 };
90
91 // transforms a FeatureCollection to an array of geoJSON objects
92 function input2array( geojsondata) {
93     // check if geojsondata is a FeatureCollection
94     if (geojsondata.type == "FeatureCollection") {
95         //console.log("mod_m_simple: detected FeatureCollection");
96         return(geojsondata.features);
97     } else if (geojsondata.type == "Feature") {
98         //console.log("mod_m_simple: detected Feature");
99         // if there is only one geoJSON feature, return it as an array
100        var gtype = geojsondata.geometry.type;
101        if (gtype == "Point" || gtype == "Polygon" || gtype == "LineString") {
102            return( [geojsondata] );
103        } else {
104            //console.log("mod_m_simple: detected unsupported Type: " + gtype);

```

```

105     // non of the above, nothing to do
106     return([]);
107 };
108 } else {
109     console.log("mod_m_simple: detected unsupported Type: " + geojsondata.type);
110     return([]);
111 };
112 };
113
114 // builds a geoJSON polygon describing a box around a point
115 function buildGjsonBox(point,extent){
116     cPoint = point.geometry.coordinates;
117     if(point.geometry.type == "LineString"){
118         var gjsonBox = {type : "Feature",
119             geometry : { type : "Polygon", coordinates : [[]] }
120         };
121
122         var allx = [];
123         var ally = [];
124         for(var i=0;i<cPoint.length;i++){
125             allx.push(cPoint[i][0]);
126             ally.push(cPoint[i][1]);
127         };
128         var top = getMax(ally) + extent;
129         var bottom = getMin(ally) - extent;
130         var left = getMin(allx) - extent;
131         var right = getMax(allx) + extent;
132
133     } else if(point.geometry.type == "Point"){
134         var top = cPoint[1] + extent;
135         var bottom = cPoint[1] - extent;
136         var left = cPoint[0] - extent;
137         var right = cPoint[0] + extent;
138     } else {
139         console.log("Unsupported geometry type: " + point.geometry.type);
140         var top = 0;
141         var bottom = 0;
142         var left = 0;
143         var right = 0;
144     };
145     var gjsonBox = {type : "Feature",
146         geometry : {type : "Polygon", coordinates : [[ [left,bottom],
147             [right,bottom],
148             [right,top],
149             [left,top],
150             [left,bottom]
151             ] ] }
152     };
153     return(gjsonBox);
154 };
155
156
157 // builds a geoJSON polygon describing a circle around a point
158 function buildRadius(point,extent){
159     // a radius search makes only sense when the element is a single point
160     // this function is solely for generating a visual representation
161     if (point.geometry.type == "Point"){

```

```
162     var x = point.geometry.coordinates[0];
163     var y = point.geometry.coordinates[1];
164     var circleElement = {type : "Feature",
165                         geometry : { type : "Polygon", coordinates : [[]]}
166                             };
167
168     // loop until the circle is complete
169     for (var i=0;i<=15;i++){
170         var circlePosition = (i * 22.5) * Math.PI / 180;
171         var singleCoord = [];
172         singleCoord.push(x+extent*Math.cos(circlePosition));
173         singleCoord.push(y+extent*Math.sin(circlePosition));
174         circleElement.geometry.coordinates[0].push(singleCoord);
175     };
176     // the first coordinate needs to be the last for the polygon to be valid
177     var singleCoord = [];
178     singleCoord.push(x+extent*Math.cos(0));
179     singleCoord.push(y+extent*Math.sin(0));
180     circleElement.geometry.coordinates[0].push(singleCoord);
181     return(circleElement);
182 } else {
183     // if the geometry is no point, use the default bounding box
184     return( buildGjsonBox(point,extent) );
185 };
186 };
187
188 // get the smallest entry of an array
189 function getMin(inarray){
190     var outvalue = inarray[0];
191     for(var i=0;i<inarray.length;i++){
192         if(outvalue > inarray[i]){outvalue = inarray[i]};
193     }
194     return(outvalue);
195 };
196
197 // get the biggest entry of an array
198 function getMax(inarray){
199     var outvalue = inarray[0];
200     for(var i=0;i<inarray.length;i++){
201         if(outvalue < inarray[i]){outvalue = inarray[i]};
202     }
203     return(outvalue);
204 };
205
206 }) ();
```

Source Code of the Web Worker

```
1  var statuslen = 0;
2  var checkelement;
3  var finalcheckhit = [];
4  var finalcheckmiss = [];
5  var finalcheckhit_c = [];
6  var alreadymatched_ids = [];
7
8  // if the worker receives a message from the parent web-application
```

```
9 onmessage = function(message) {
10   consolelog(" --- init the mod_m_simple WebWorker");
11
12   if (message.data.command == "start") {
13     var checklen = message.data.content.length;
14     var connectionstring = message.data.connection;
15     var checkarray = message.data.content;
16     var checksettings = message.data.checksettings;
17
18     // iterate over all elements and perform check
19     doiterate = 1;
20     alreadymatched_ids = [];
21     for (var i=0;i<checklen;i++){
22       statuslen = i;
23       checkelement = checkarray[i];
24
25       // for every 1-percent-change post a status update
26       if ( (Math.floor(100/checklen*statuslen)) % 1 == 0) {
27         postStatus(Math.floor(100/checklen*statuslen));
28       }
29
30       // generate the query that has to be performed
31       var sradius = checksettings.searchradius;
32
33       var query = generateQuery(checkelement,connectionstring,sradius);
34       //consolelog(query);
35       // execute the query and gather its results
36       var resultlist = doQuery(query).rows;
37       // clean the resultlist from elements that were already matched
38       resultlist = removeAlreadyChecked(resultlist,alreadymatched_ids);
39       // evaluate the results
40       if (resultlist.length > 0){
41         var result = postProcess(checkelement, resultlist);
42         checkelement.checkdistance = result.properties.checkdistance;
43
44         // if specified, do a radius search
45         if(checksettings.radiussearch == true &&
46           checkelement.geometry.type == "Point" &&
47           checkelement.checkdistance > checksettings.searchradius){
48           consolelog("IN BBOX BUT NOT IN RADIUS: "+checkelement._id);
49           checkelement.layer = "miss";
50           finalcheckmiss.push(checkelement);
51           pushToMap(checkelement);
52           continue;
53         };
54         checkelement.layer = "hit";
55         checkelement.partnerId = result._id;
56         result.partnerId = checkelement._id;
57         result.checkdistance = checkelement.checkdistance;
58         result.layer = "hitc";
59         alreadymatched_ids.push(result._id);
60         finalcheckhit_c.push(result);
61         pushToMap(checkelement);
62         finalcheckhit.push(checkelement);
63         // next the line connecting the matching-pair is drawn
64         if(checkelement.geometry.type == "Point"){
65           var linefeat = getEmptyGeoJSON("LineString");
```

```

66         linefeat.geometry.coordinates = [checkelement.geometry.coordinates,
67                                         result.geometry.coordinates];
68     } else {
69         var linefeat = getEmptyGeoJSON("LineString");
70         linefeat.geometry.coordinates = [checkelement.geometry.coordinates[0],
71                                         result.geometry.coordinates];
72     };
73     pushToMap( linefeat );
74
75     } else {
76         // when there is no result, the check is a MISS
77         checkelement.layer = "miss";
78         finalcheckmiss.push(checkelement);
79         pushToMap(checkelement);
80     }
81 }; //loop over checkarray
82 doiterate = 0;
83 postMessage({command: "finished",
84             content: { hit: finalcheckhit,
85                     miss: finalcheckmiss,
86                     hitc: finalcheckhit_c,
87             }});
88 self.close();
89
90 } else if (message.data.command == "stop") {
91     consolelog("got command to stop worker");
92     self.close();
93
94 } else if (message.data.command == "status") {
95     postStatus(Math.floor(100/checklen*statuslen));
96 };
97
98 // elements with their id in a list are removed from the 'resultlist'
99 function removeAlreadyChecked(resultlist,alreadymatchedlist){
100     var newresultlist = [];
101     for(var i=0;i<resultlist.length;i++){
102         var singleresult = resultlist[i];
103         if(alreadymatchedlist.indexOf(singleresult.id) > -1){
104             // this entry has already been checked
105         }else{
106             // this entry has not yet been checked
107             newresultlist.push(singleresult);
108         };
109     };
110     return(newresultlist);
111 };
112
113 // creates an empty geoJSON document
114 function getEmptyGeoJSON(geojsonType){
115     var geoJSON = {type: "Feature",
116                   layer: "checkdata",
117                   geometry : { type: "LineString",
118                               coordinates: []
119                   }
120     };
121     geoJSON.geometry.type = geojsonType;
122     return(geoJSON);

```

```
123     };
124
125     // sends a geoJSON to the map
126     function pushToMap(element) {
127         postMessage( {command : "addToMap", content : element} );
128     };
129
130     // a debugging function for output to the console
131     function consolelog(infos) {
132         postMessage( {"command" : "info", content : infos} );
133     }
134
135     // for status-updates
136     function postStatus(stat) {
137         postMessage( {command : "status", content : stat} );
138     }
139
140     // create a http-query for a couchdb-boundingbox request
141     function generateQuery(element,connectionstring,searchradius) {
142         var bbox = buildBbox(element, searchradius);
143         var query = connectionstring + "?bbox=" + bbox;
144         return(query);
145     }
146
147     // create a bounding-box of a certain extent around a geoJSON-feature
148     function buildBbox(element,extent) {
149         // "element" is GeoJSON
150         // return is an array of coordinates
151         var top,bottom,left,right;
152         var ecoord = element.geometry.coordinates;
153
154         if (element.geometry.type == "Point"){
155             // with a single point, it is simple to create the bounding-box
156             top = ecoord[1] + extent;
157             bottom = ecoord[1] - extent;
158             left = ecoord[0] - extent;
159             right = ecoord[0] + extent;
160         } else if (element.geometry.type == "LineString" ) {
161             // with a linestring, take the most outer values, apply the extent
162             // and create the bounding-box with these
163             var allx = [];
164             var ally = [];
165             for(var i=0;i<ecoord.length;i++){
166                 allx.push(ecoord[i][0]);
167                 ally.push(ecoord[i][1]);
168             };
169             top = getMax(ally) + extent;
170             bottom = getMin(ally) - extent;
171             left = getMin(allx) - extent;
172             right = getMax(allx) + extent;
173
174         } else {
175             consolelog("unsupported type: " + element.geometry.type);
176             top = 0;
177             bottom = 0;
178             left = 0;
179             right = 0;
```

```
180     };
181     var bbox = [left , bottom , right , top];
182     return (bbox);
183 };
184
185 // performs a query AJAX-style and returns JSON
186 function doQuery(query) {
187     var request = createAJAX();
188     request.open("GET", query, false);
189     request.send(null);
190     var response = request.responseText;
191     delete request;
192     return( eval("(" + response + ")"));
193     // return(request.responseText);
194 }
195
196 // creates an AJAX object
197 function createAJAX() {
198     return new XMLHttpRequest();
199 }
200
201 // from multiple results refine and return just one
202 function postProcess(checkelement, resultlist) {
203     var posresult;
204     if(checkelement.geometry.type == "Point"){
205         var fromPoint = checkelement.geometry.coordinates;
206     } else if (checkelement.geometry.type == "LineString"){
207         //var fromPoint = checkelement.geometry.coordinates[0];
208         var fromPoint = getCenterPoint(checkelement);
209     } else if (checkelement.geometry.type == "Polygon"){
210         var fromPoint = checkelement.geometry.coordinates[0][0];
211     };
212
213     // check every element of the resultlist
214     for (var i=0;i<resultlist.length;i++){
215         var currentelement = resultlist[i].value;
216         // calculate the distance for every element
217         var toPoint = currentelement.geometry.coordinates;
218         currentelement.properties.checkdistance = calcDistance(fromPoint,toPoint);
219         // overwrite the "key-value" entry in the array with just the modified "value"
220         resultlist[i] = currentelement;
221     };
222     // determine the nearest result
223     posresult = getMinInObject(resultlist,"checkdistance");
224     return(posresult);
225 }
226
227 //calculates the average coordinates of an array of points
228 function getCenterPoint(element){
229     var sumx = 0;
230     var sumy = 0;
231     for(var i=0;i<element.geometry.coordinates.length;i++){
232         sumx += element.geometry.coordinates[i][0];
233         sumy += element.geometry.coordinates[i][1];
234     };
235     var newx = sumx / element.geometry.coordinates.length;
236     var newy = sumy / element.geometry.coordinates.length;
```

```
237     // pushToMap({type:"Feature",geometry:{type:"Point",coordinates:[newx,newy]}});
238     return ([newx,newy]);
239 };
240
241 // get smallest value of array
242 function getMin(inarray){
243     var outvalue = inarray[0];
244     for(var i=0;i<inarray.length;i++){
245         if(outvalue > inarray[i]){outvalue = inarray[i]};
246     }
247     return(outvalue);
248 }
249
250 // get biggest value of array
251 function getMax(inarray){
252     var outvalue = inarray[0];
253     for(var i=0;i<inarray.length;i++){
254         if(outvalue < inarray[i]){outvalue = inarray[i]};
255     }
256     return(outvalue);
257 }
258
259 // get object of an array with smallest entry in property 'dfield'
260 function getMinInObject(inarray,dfield){
261     var result = inarray[0];
262     for(var i=0;i<inarray.length;i++){
263         if (result.properties[dfield] > inarray[i].properties[dfield]){
264             result = inarray[i];
265         }
266     }
267     return(result);
268 }
269
270 // calculates plain distance between two pairs of coordinates
271 function calcDistance(p1,p2){
272     // convert to radiants
273     lon1 = p1[0] * Math.PI / 180;
274     lat1 = p1[1] * Math.PI / 180;
275     lon2 = p2[0] * Math.PI / 180;
276     lat2 = p2[1] * Math.PI / 180;
277
278     var x = (lon2-lon1) * Math.cos((lat1+lat2)/2);
279     var y = (lat2-lat1);
280     // var R = 6371000; // Radius of the earth in m
281     // var d = Math.sqrt(x*x + y*y) * R; // necessary if distance wanted in m
282     var d = Math.sqrt(x*x + y*y)
283
284     // convert radiants back to degrees
285     d = d * 180 / Math.PI;
286     return(d);
287 }
288 }; //onMessage
```

7.3 Comparison Maps

Comparison of the Matching Procedures - Vienna, Josefstadt

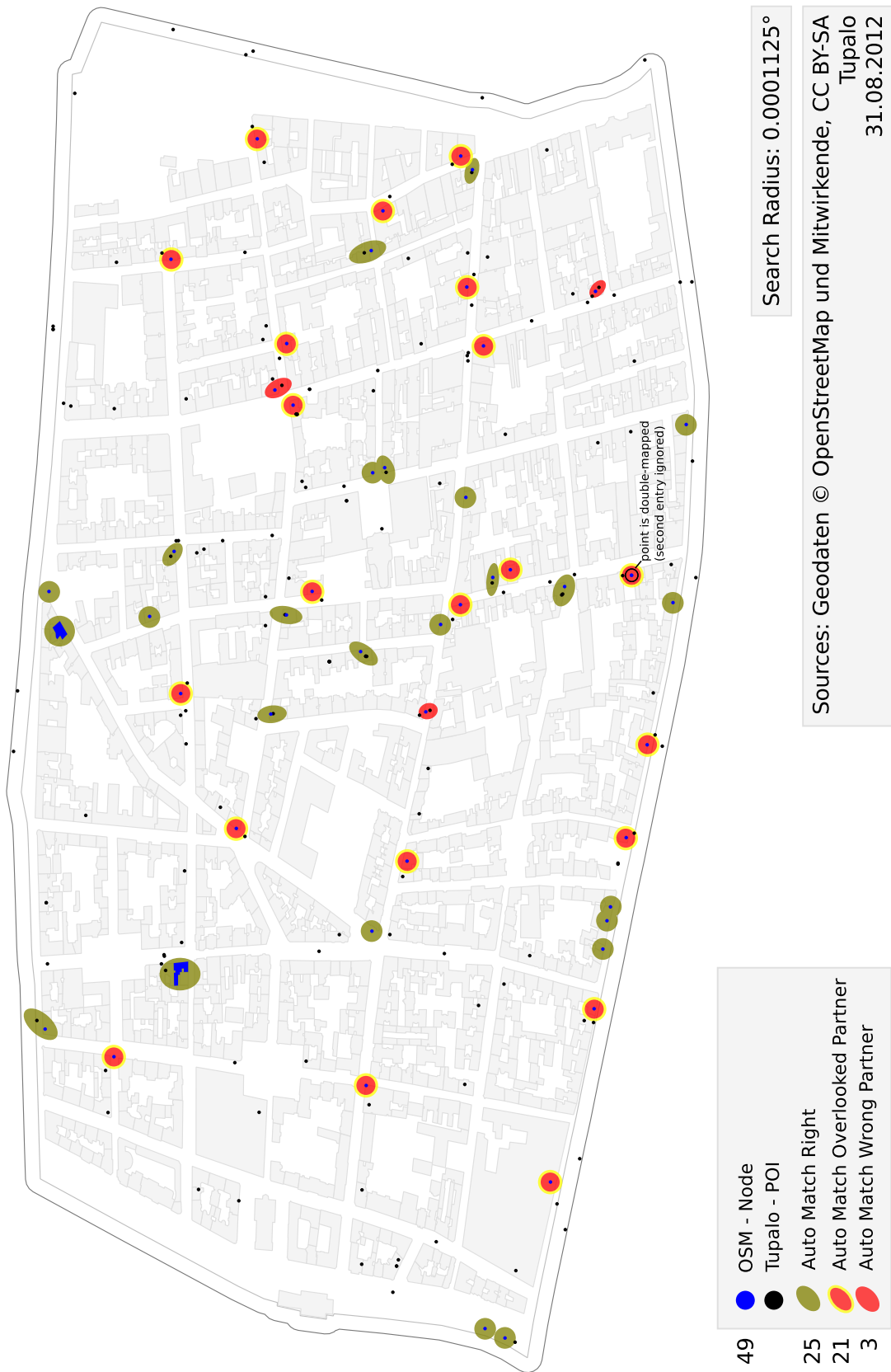


Figure 7.1: Comparison of the Matching Procedures - Vienna, Josefstadt (0.0001125°)

Comparison of the Matching Procedures - Vienna, Josefstadt

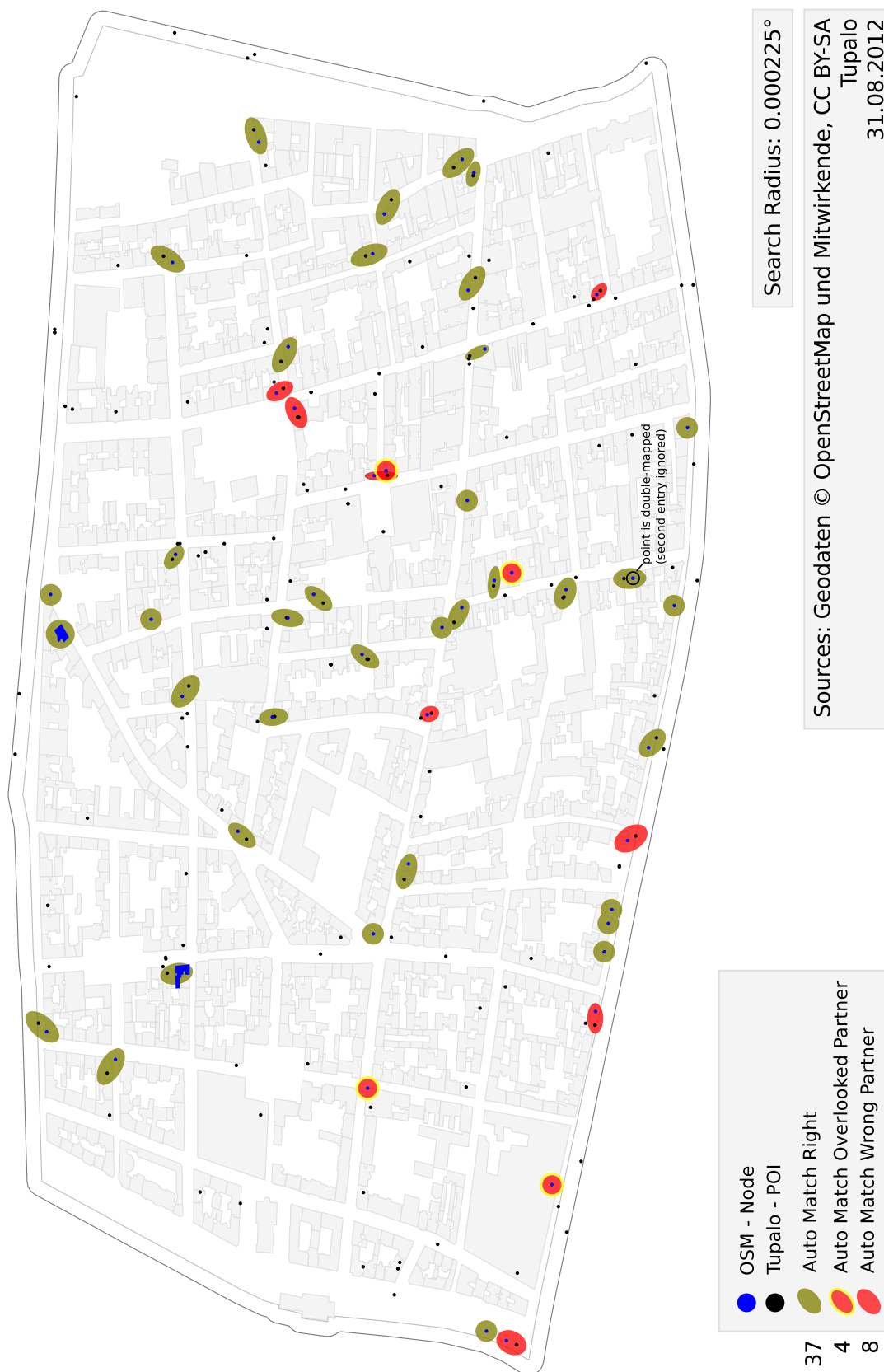


Figure 7.2: Comparison of the Matching Procedures - Vienna, Josefstadt (0.000225°)

Comparison of the Matching Procedures - Vienna, Josefstadt

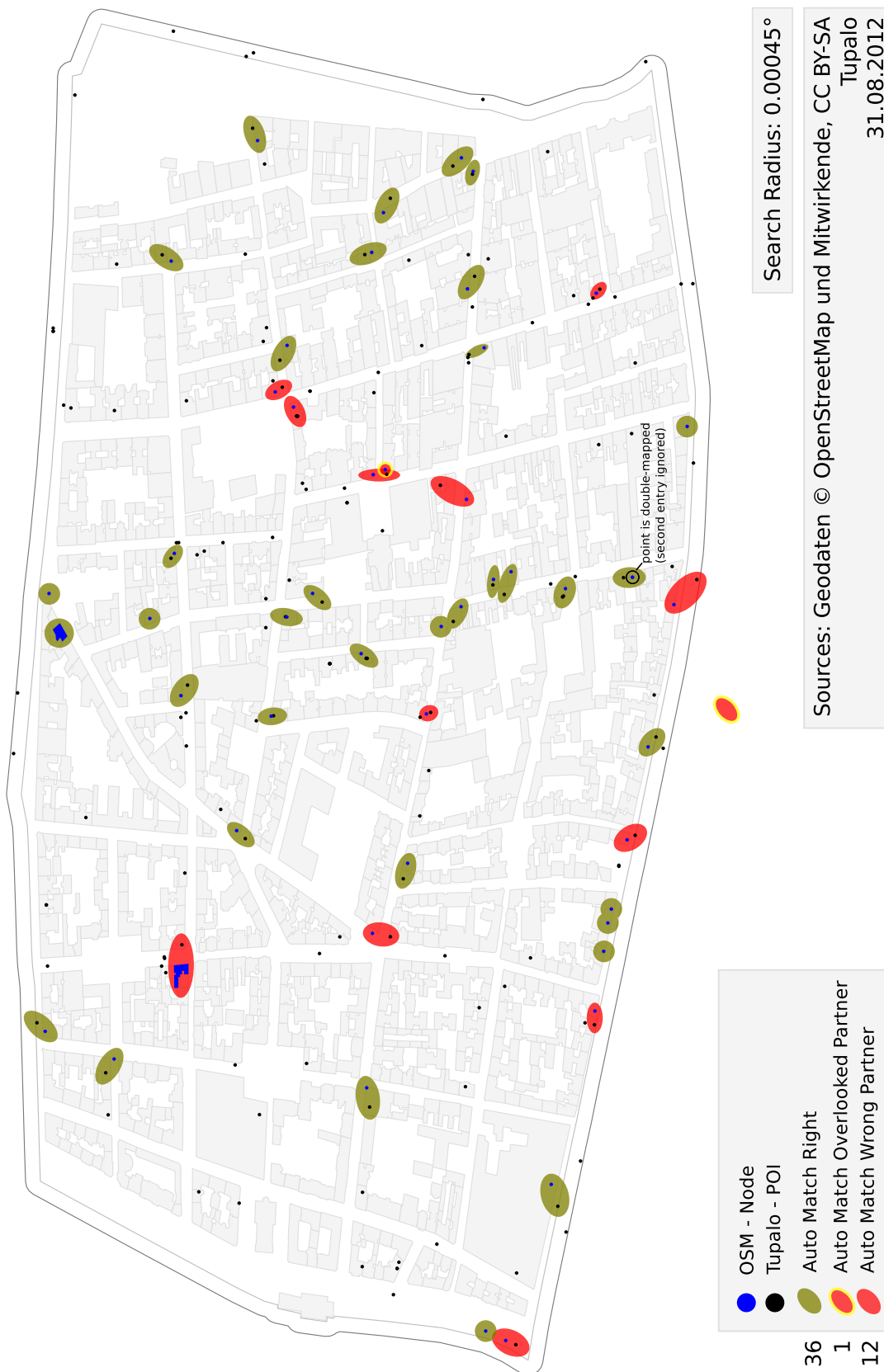


Figure 7.3: Comparison of the Matching Procedures - Vienna, Josefstadt (0.00045°)

Comparison of the Matching Procedures - Vienna, Josefstadt

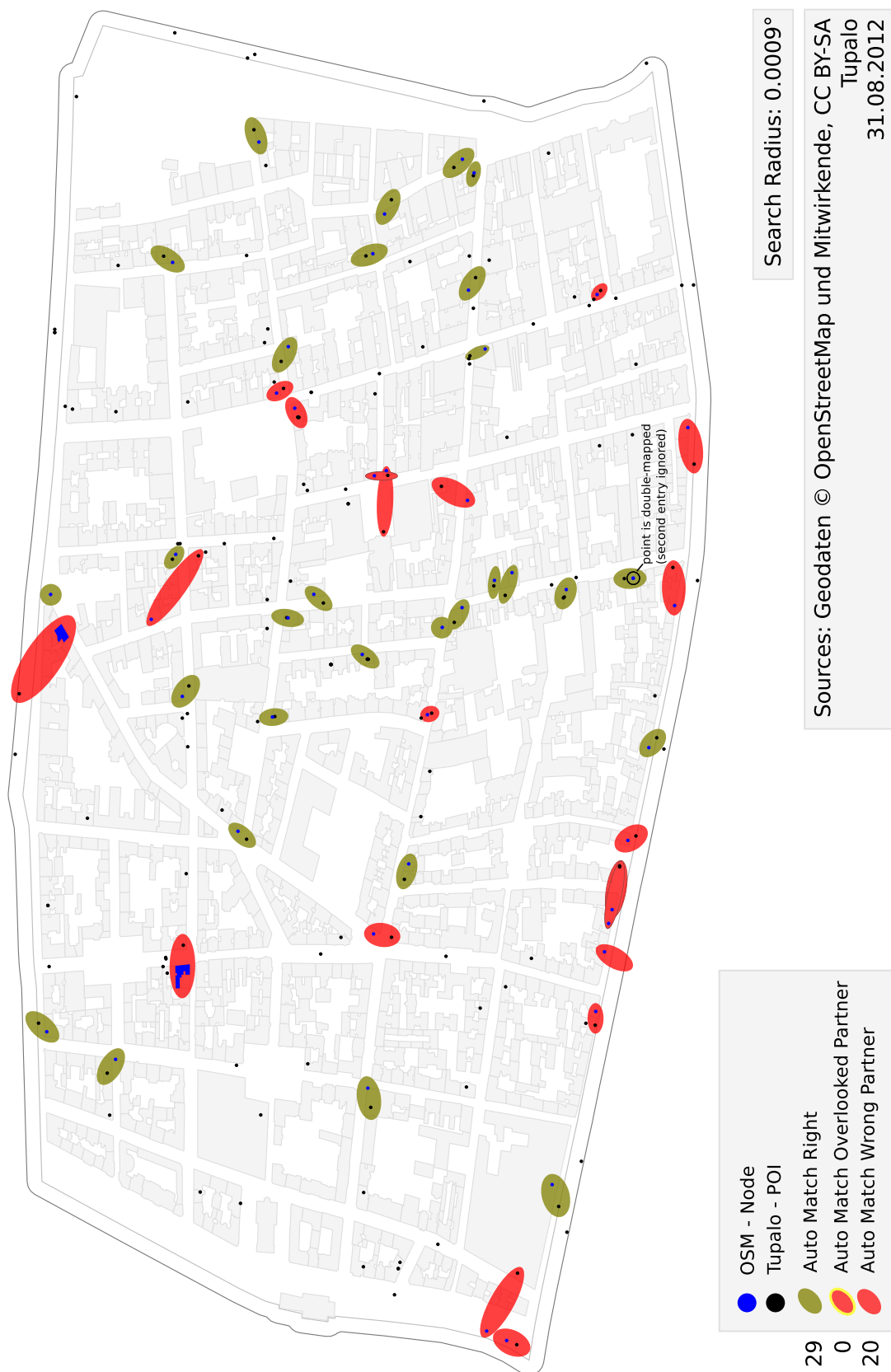


Figure 7.4: Comparison of the Matching Procedures - Vienna, Josefstadt (0.0009°)

Comparison of the Matching Procedures - Vienna, Wieden

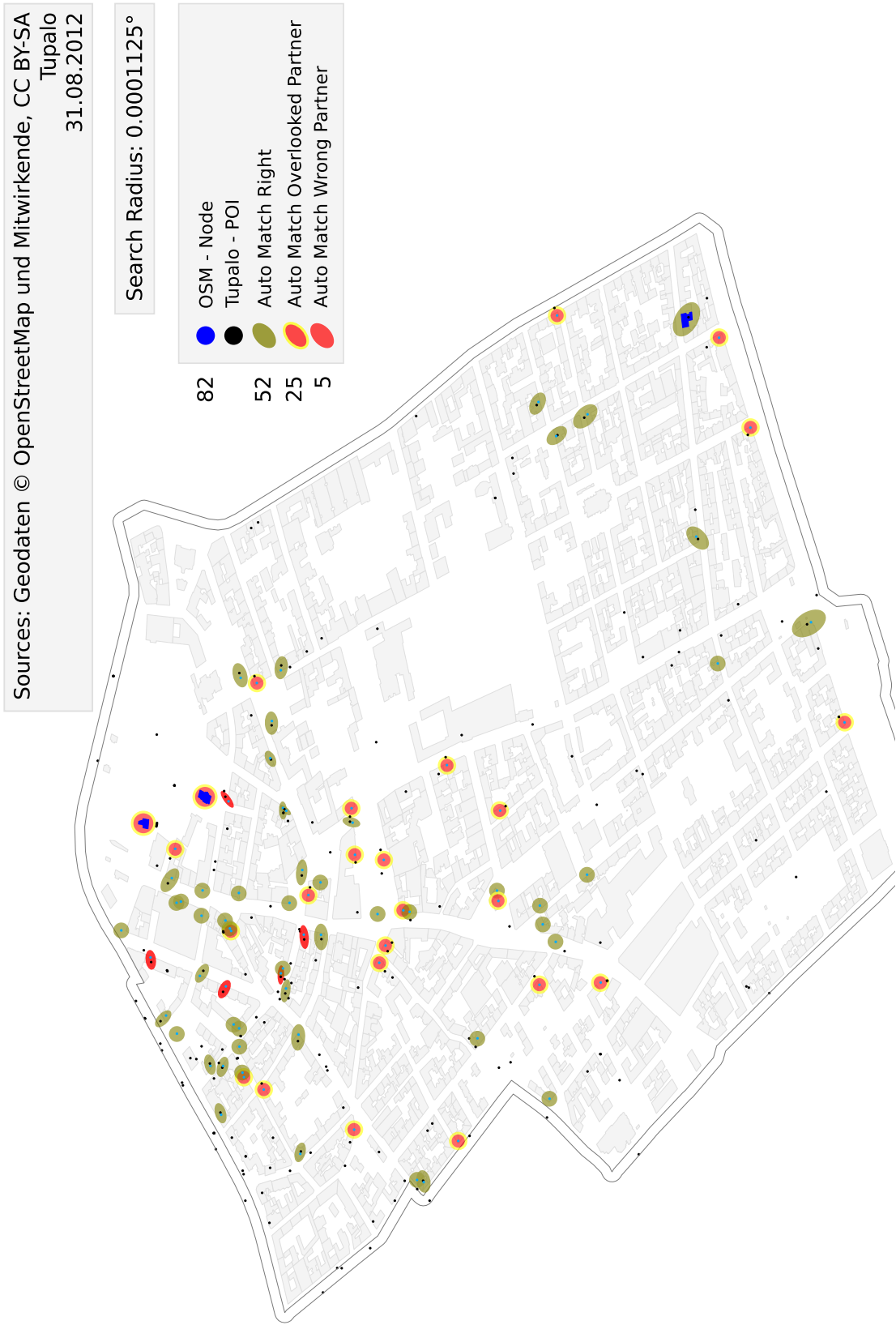


Figure 7.5: Comparison of the Matching Procedures - Vienna, Wieden (0.0001125°)

Comparison of the Matching Procedures - Vienna, Wieden

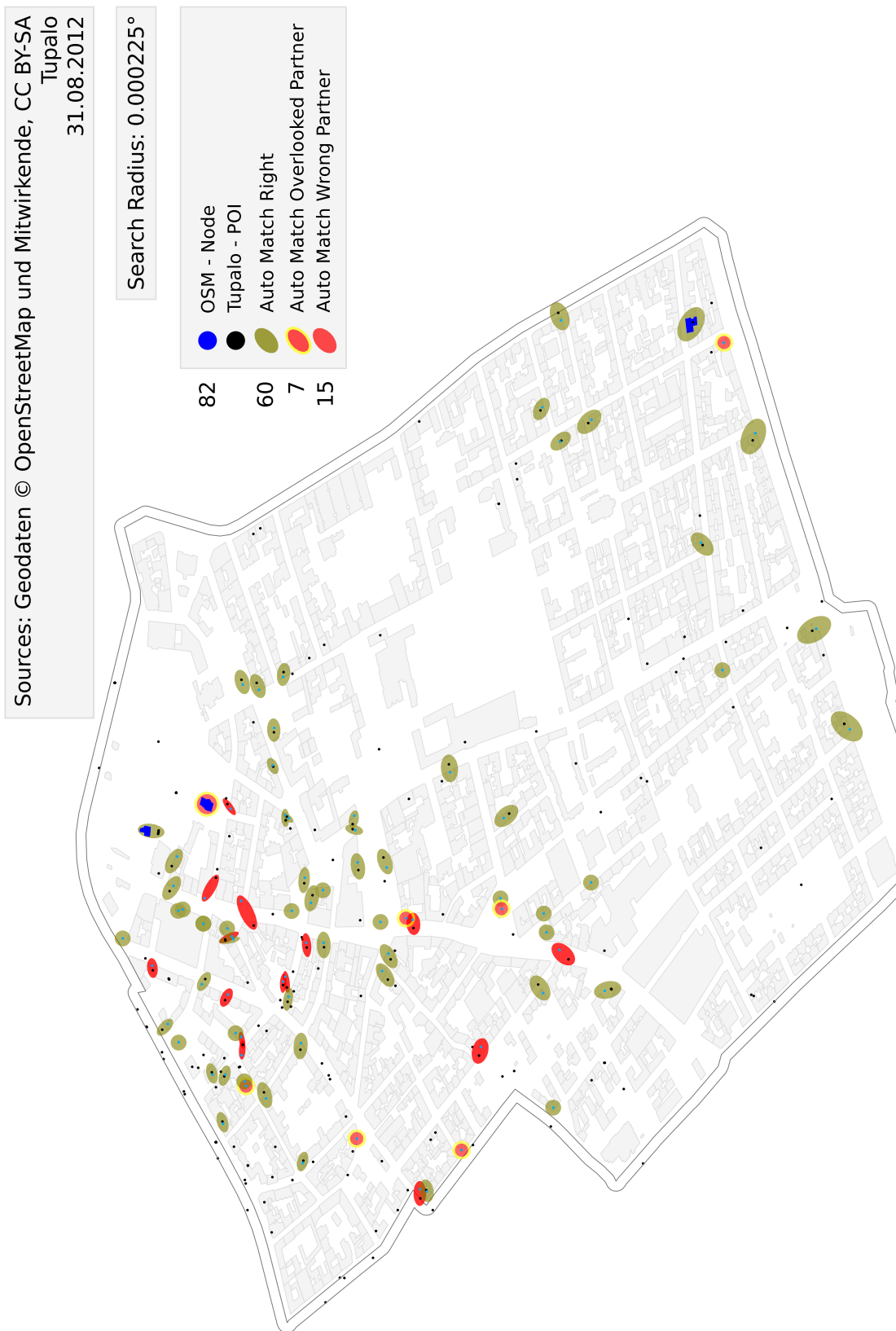


Figure 7.6: Comparison of the Matching Procedures - Vienna, Wieden (0.000225°)

Comparison of the Matching Procedures - Vienna, Wieden

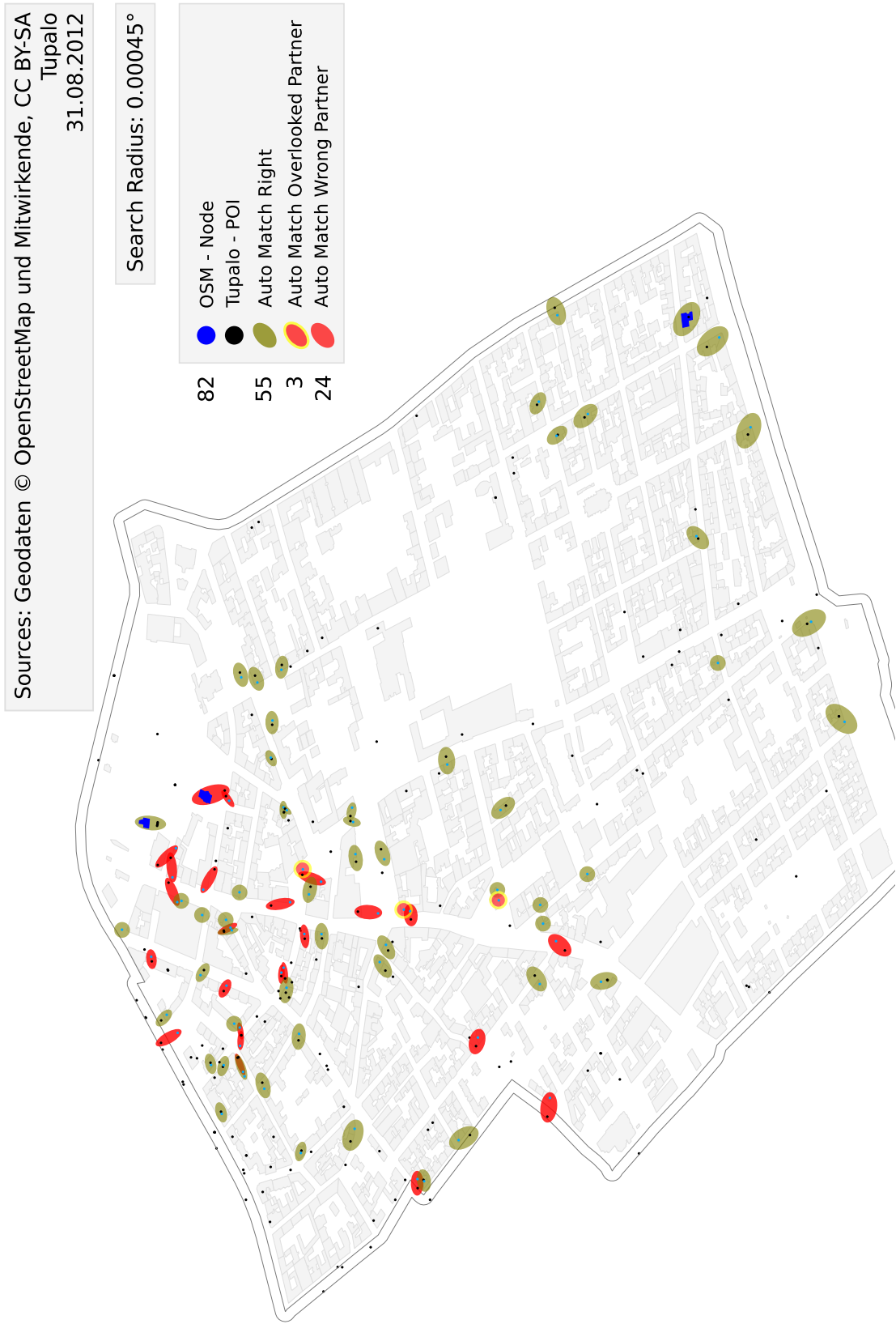


Figure 7.7: Comparison of the Matching Procedures - Vienna, Wieden (0.00045°)

Comparison of the Matching Procedures - Vienna, Wieden

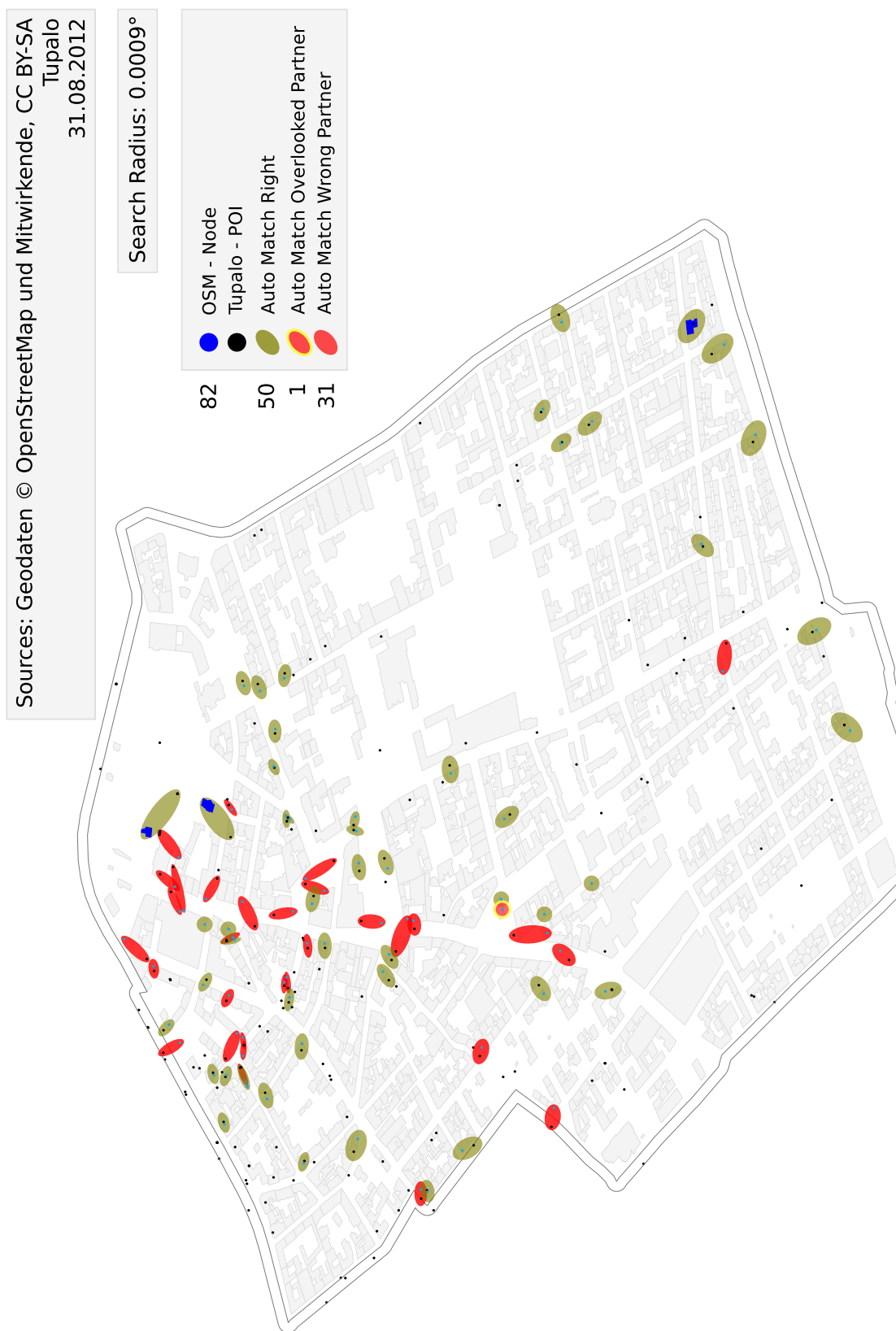


Figure 7.8: Comparison of the Matching Procedures - Vienna, Wieden (0.0009°)

Comparison of the Matching Procedures - Graz, Lend

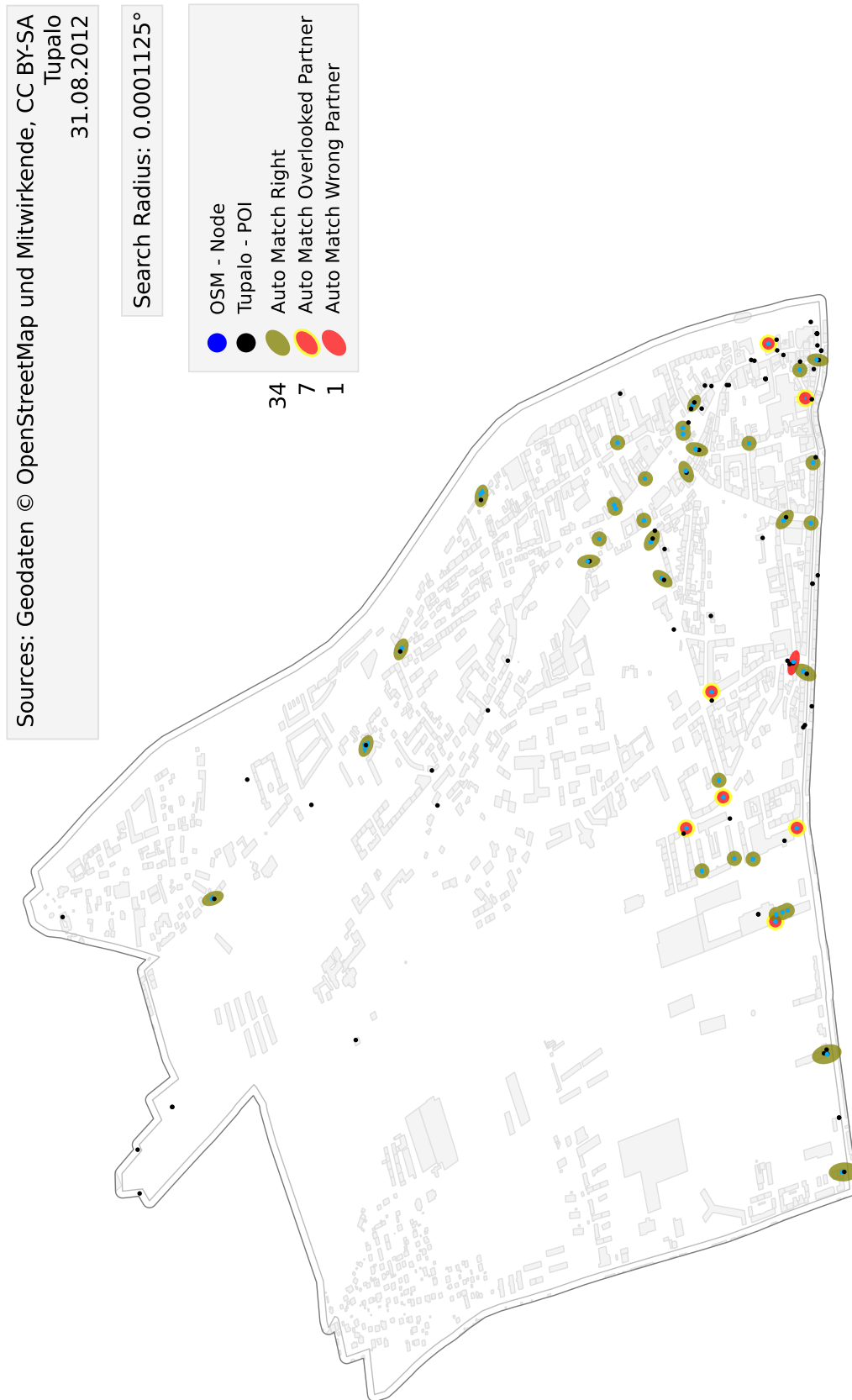


Figure 7.9: Comparison of the Matching Procedures - Graz, Lend (0.0001125°)

Comparison of the Matching Procedures - Graz, Lend

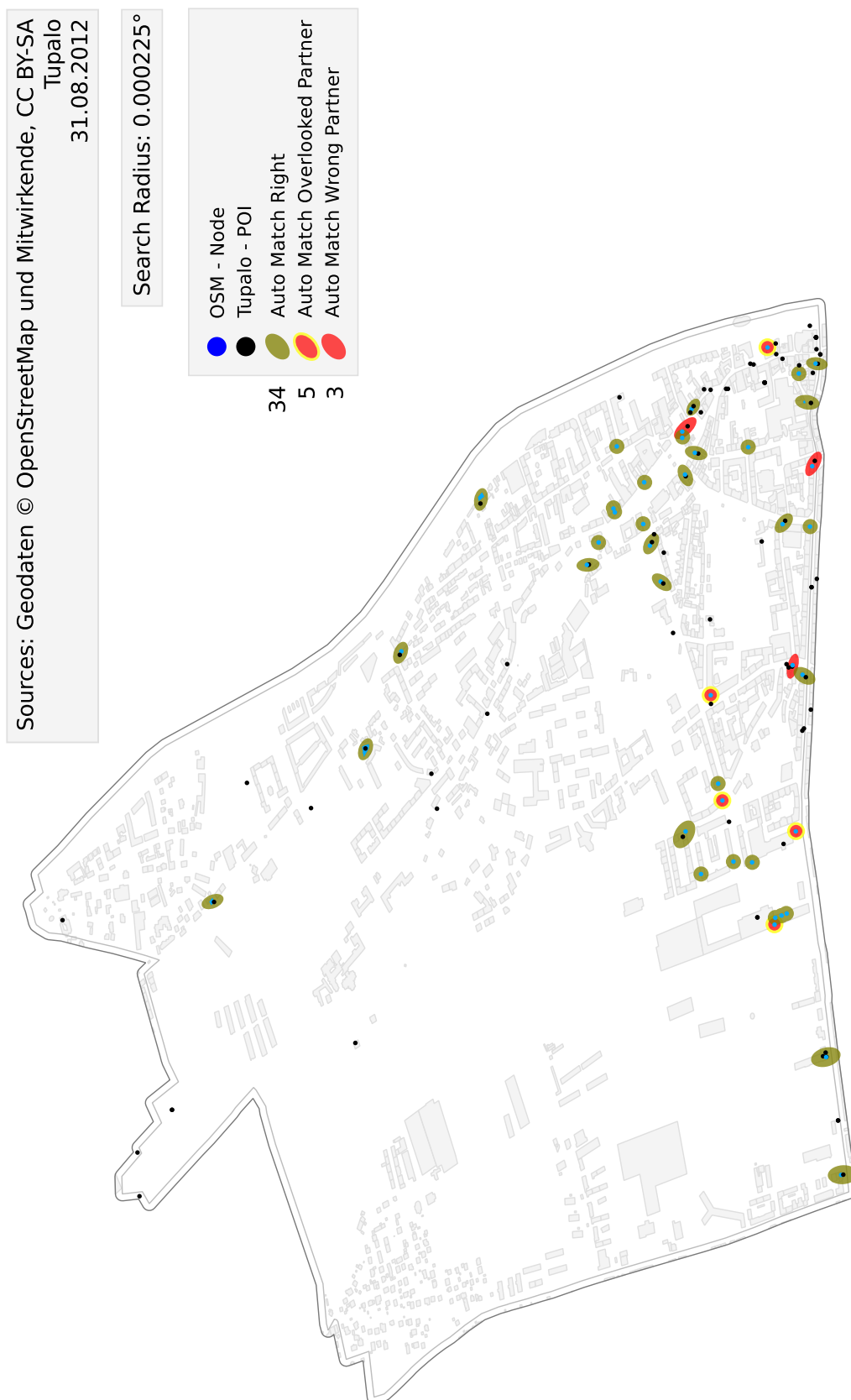


Figure 7.10: Comparison of the Matching Procedures - Graz, Lend (0.000225°)

Comparison of the Matching Procedures - Graz, Lend

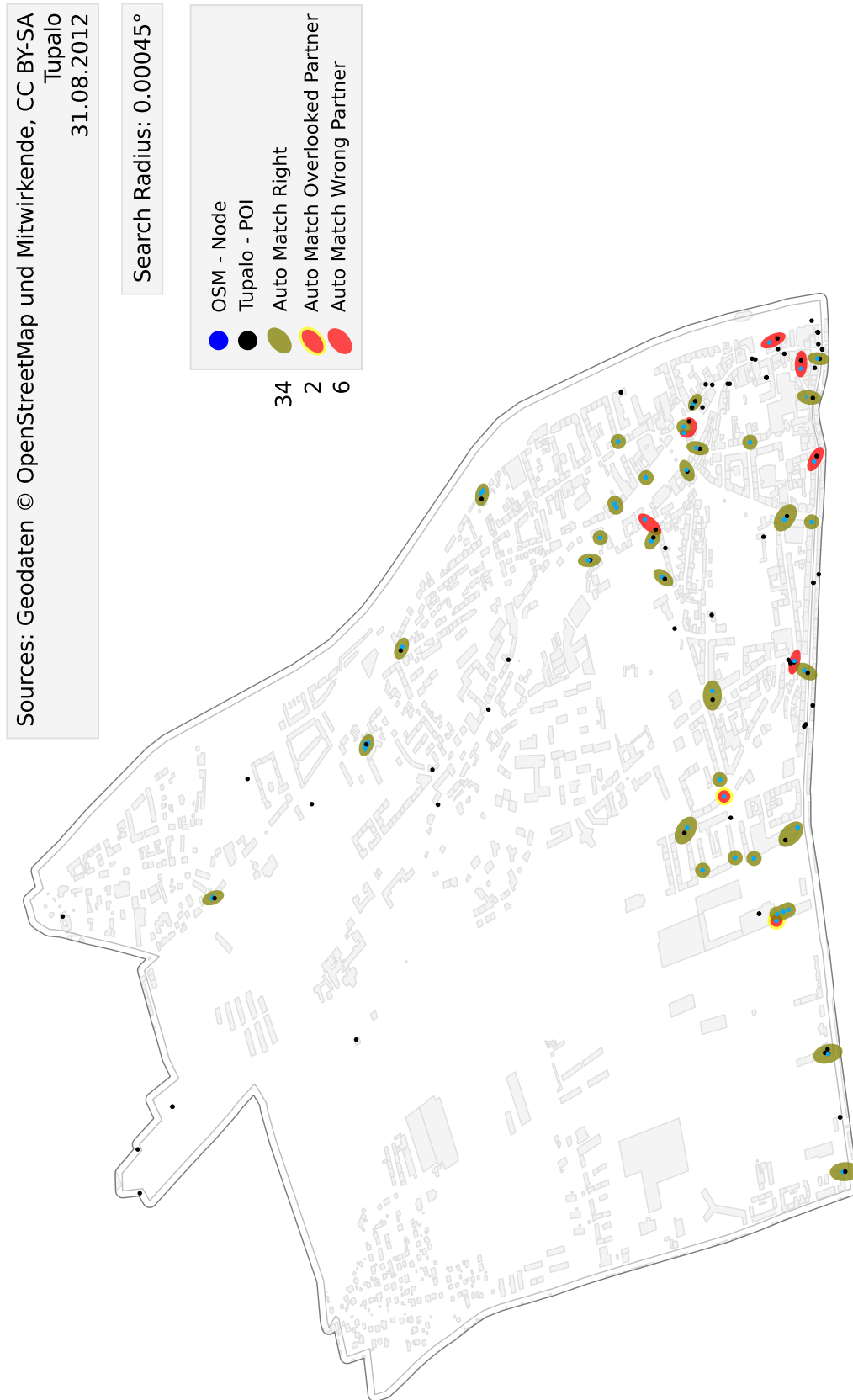


Figure 7.11: Comparison of the Matching Procedures - Graz, Lend (0.00045°)

Comparison of the Matching Procedures - Graz, Lend

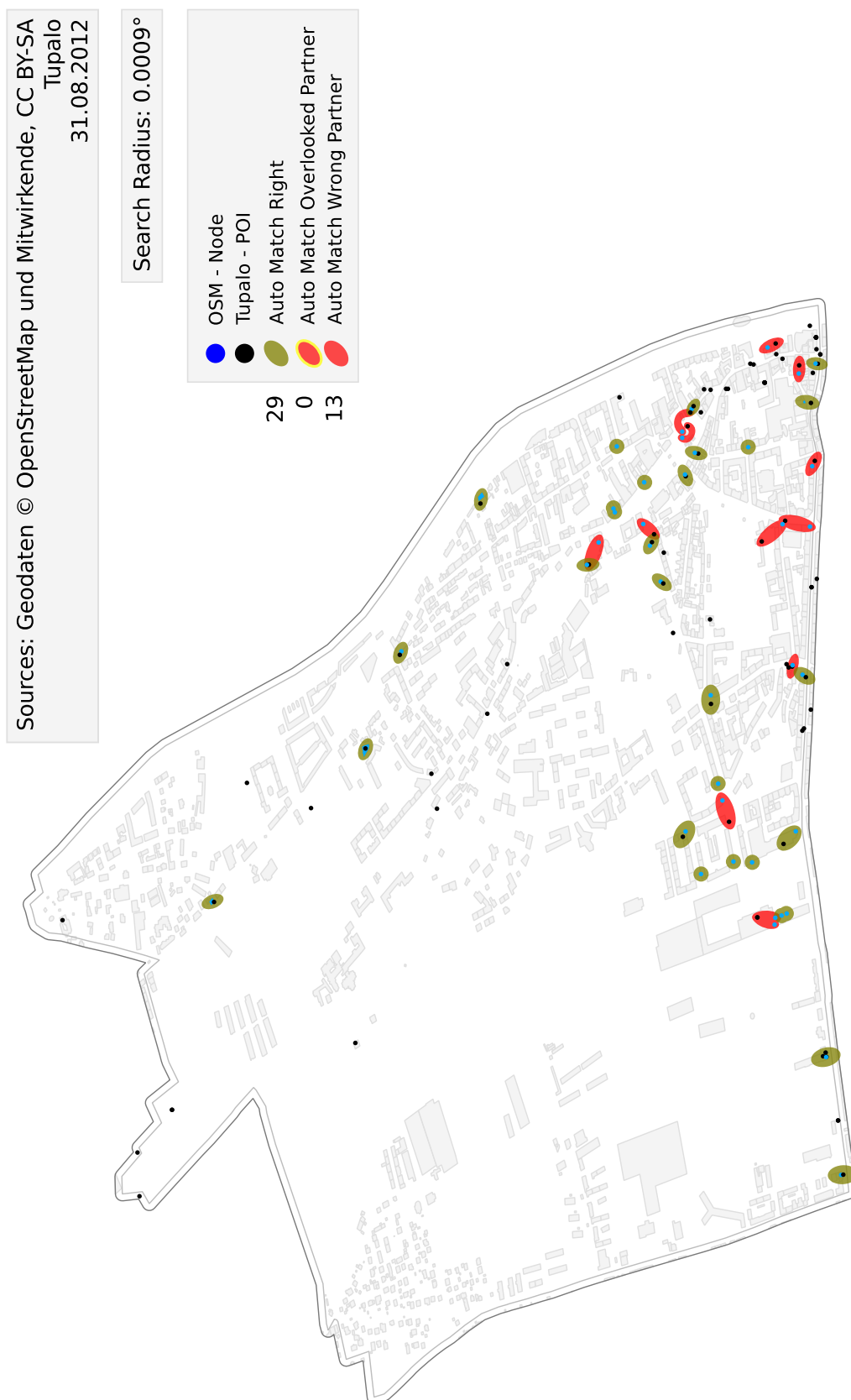


Figure 7.12: Comparison of the Matching Procedures - Graz, Lend (0.0009°)

Comparison of the Matching Procedures - Graz, Innere Stadt

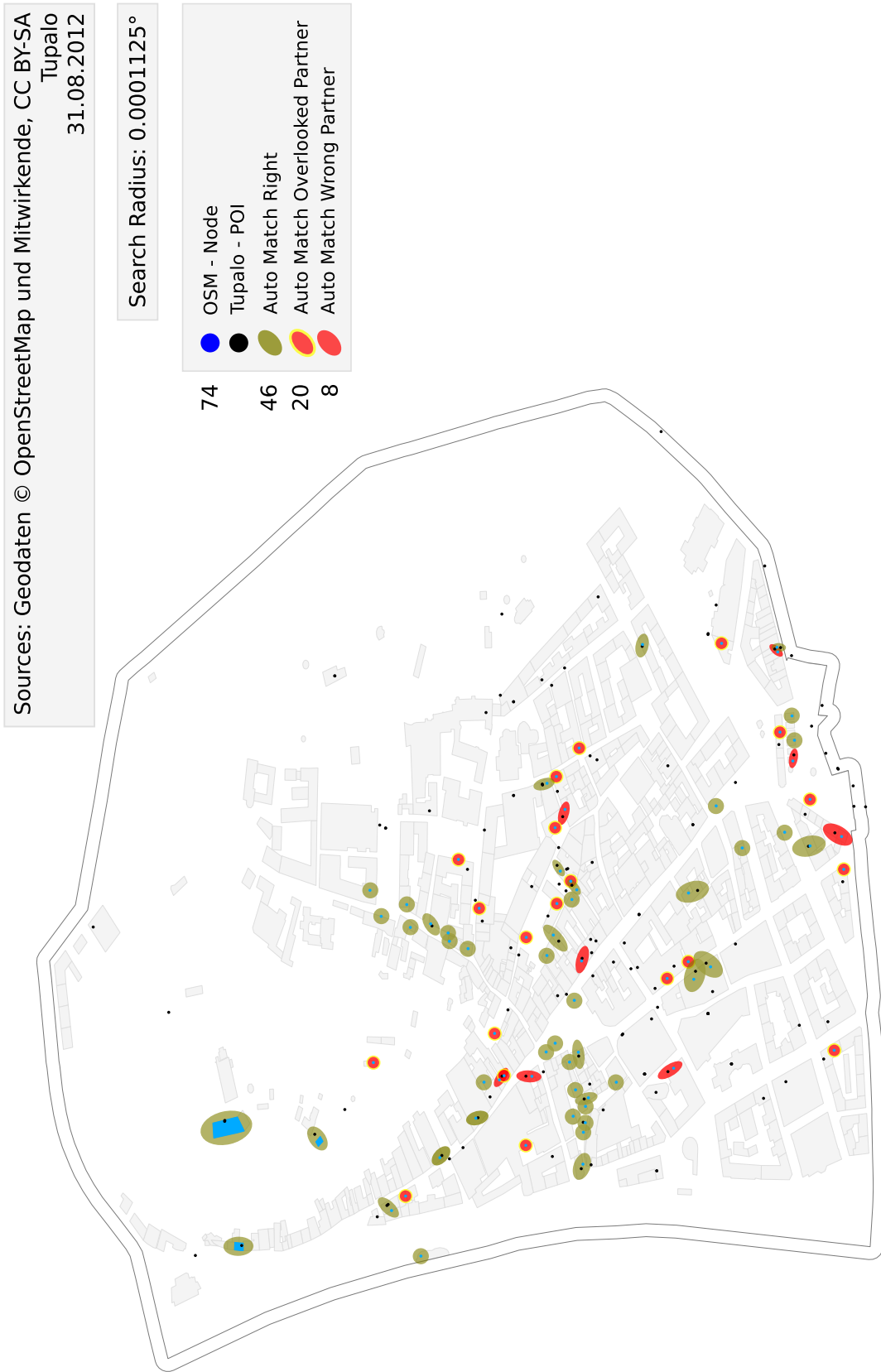


Figure 7.13: Comparison of the Matching Procedures - Graz, Innere Stadt (0.0001125°)

Comparison of the Matching Procedures - Graz, Innere Stadt

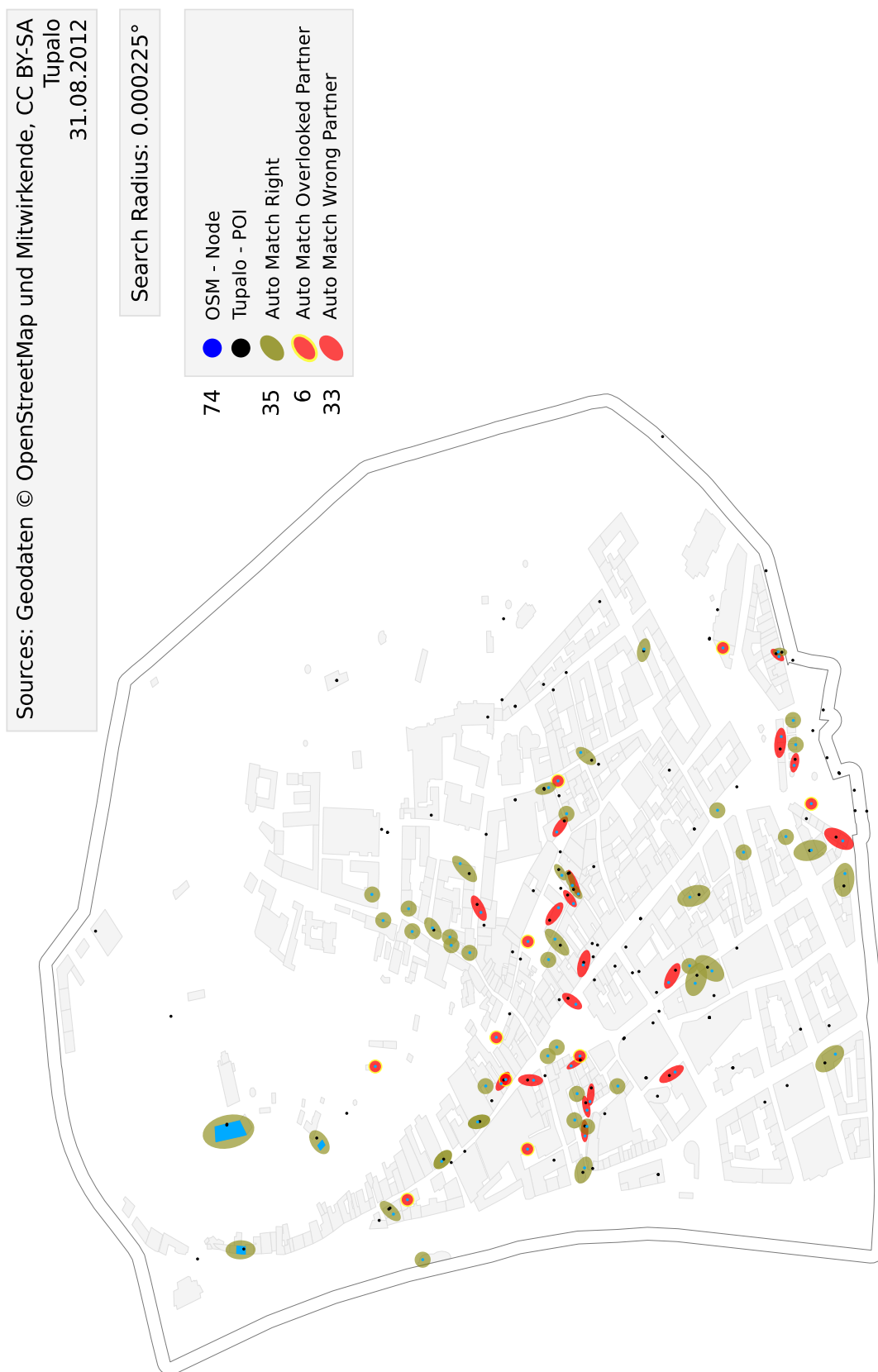


Figure 7.14: Comparison of the Matching Procedures - Graz, Innere Stadt (0.000225°)

Comparison of the Matching Procedures - Graz, Innere Stadt

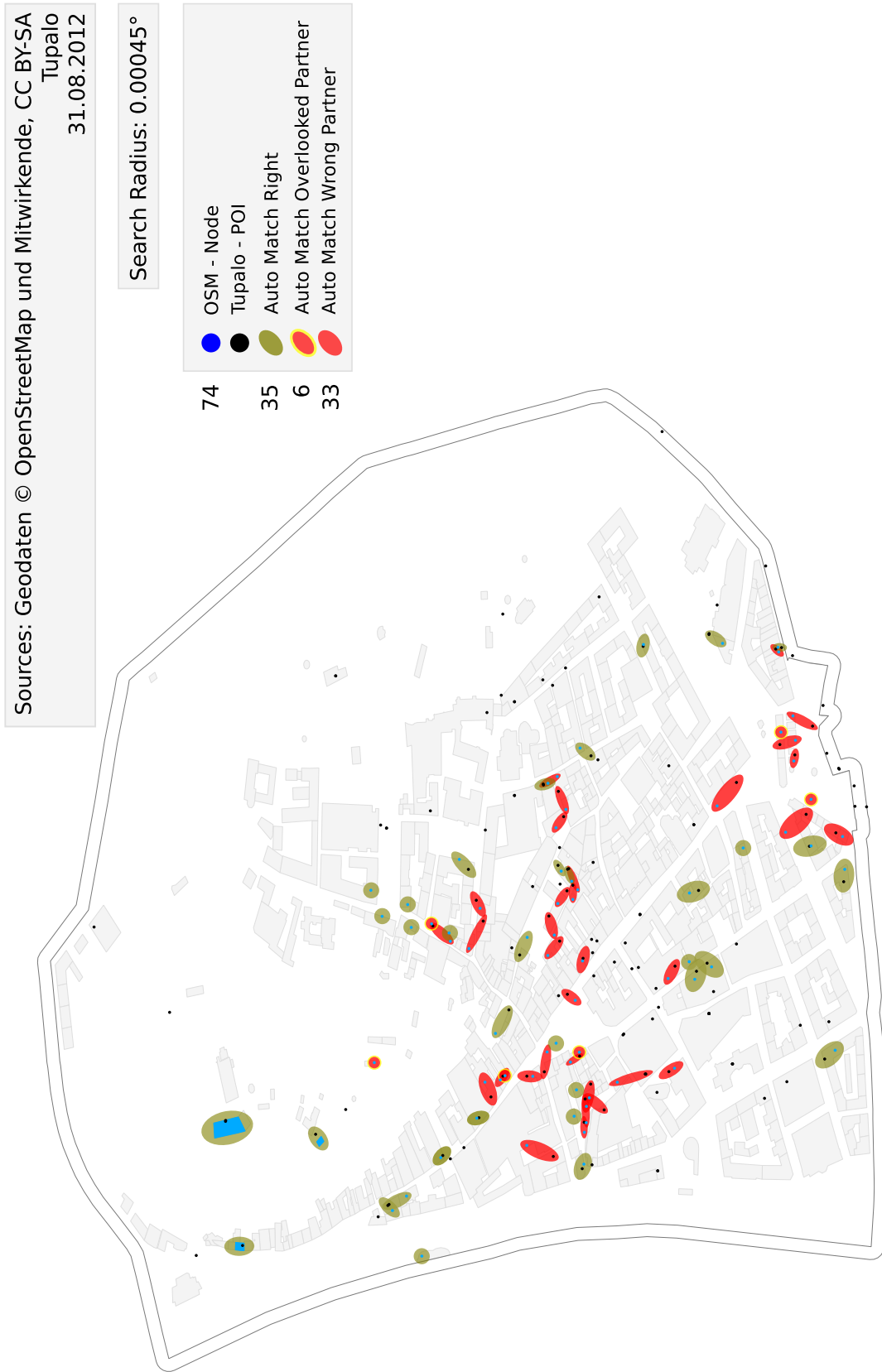


Figure 7.15: Comparison of the Matching Procedures - Graz, Innere Stadt (0.00045°)

Comparison of the Matching Procedures - Graz, Innere Stadt

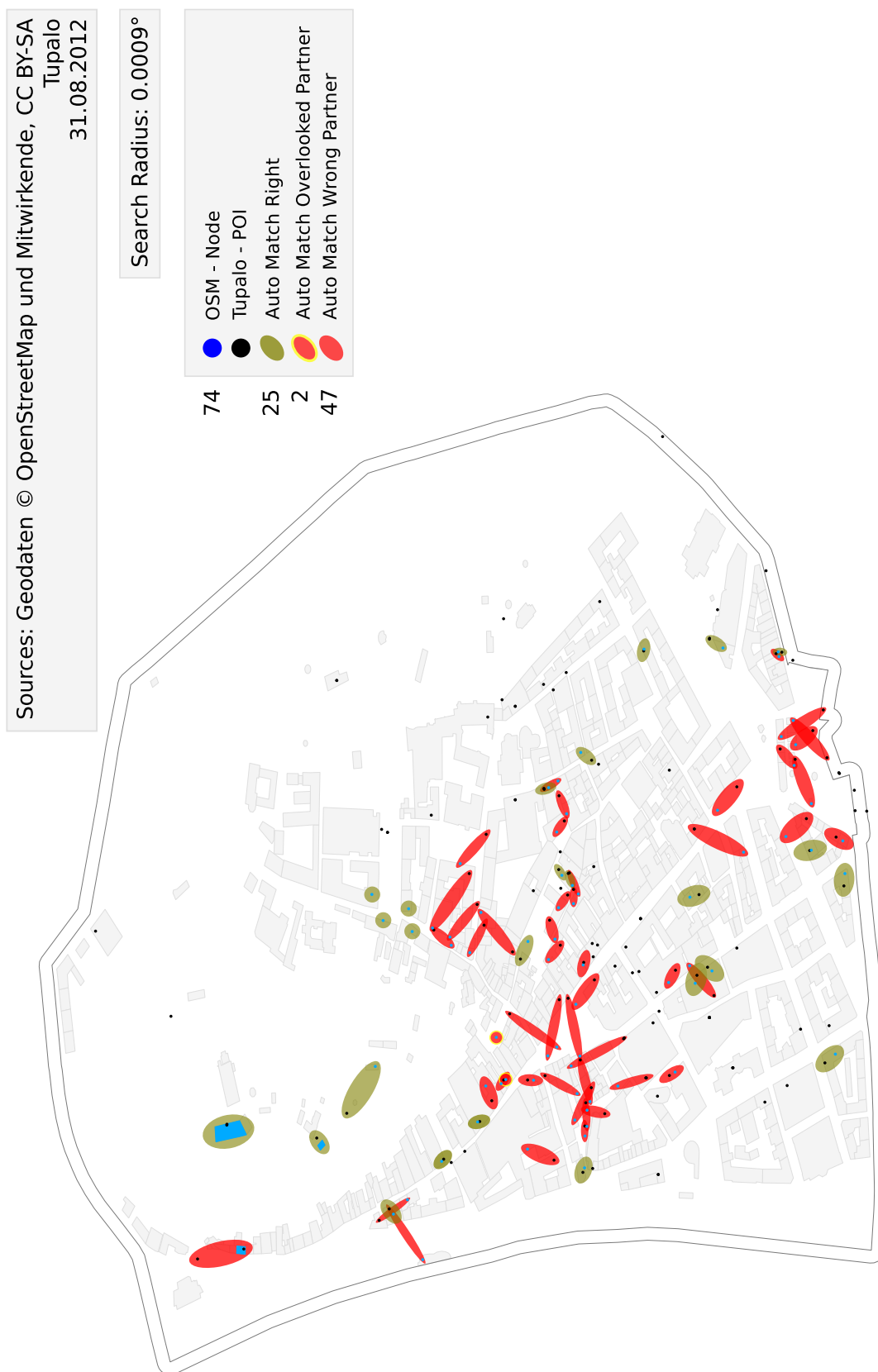


Figure 7.16: Comparison of the Matching Procedures - Graz, Innere Stadt (0.0009°)

7.4 Pictures of the Web Application

▼ (1) Define Comparative Dataset

The comparative dataset is the data which will be used to compare the OSM database to. Choose an available dataset:

baumogd x
tupalo_graz x
tupalo_wien x

Upload a new dataset:

Add a dataset

Figure 7.17: Menu for Selection of the Comparative Dataset

▼ (2) Define Categories

The category definition is the filter by which the entries of the OSM dataset are selected. It should correspond to the ontological description of your comparative dataset. Available category definitions:

natural-tree x
amenity-restaurant x

Create a new category definition:

New Selection

Figure 7.18: Menu for Matching Categories

▼ (3) Define Matching Area

The matching area is the extent you want to perform the matching on. Available boundaries:

graz x
josefstadt x
vienna x
wieden x

Define new extent:

Add an area

Figure 7.19: Menu for the Area of Matching

▼ (4) Define Matching Radius

Enter the desired matching radius in degrees:

Figure 7.20: Menu for the Definition of the Search Radius

▼ (5) Define Legal Properties

Do you want to operate in "Private Mode"?

Is the comparative dataset ODbL compatible?

Figure 7.21: Menu for defining Legal Properties

Legal Recommendations

Private Mode: **false**
Reuse the current map view: **true**
Reuse non OSM Results (CSV export): **false**
Reuse OSM Results (CSV export): **true**
Reuse the statistics: **false**

Deactivate the layer "Comparative Data" and "Detailed Check Data" to only display OSM information in the map.

The legal recommendation given here only applies in a general manner. The actual situation may be different.

Figure 7.22: Legal Recommendations

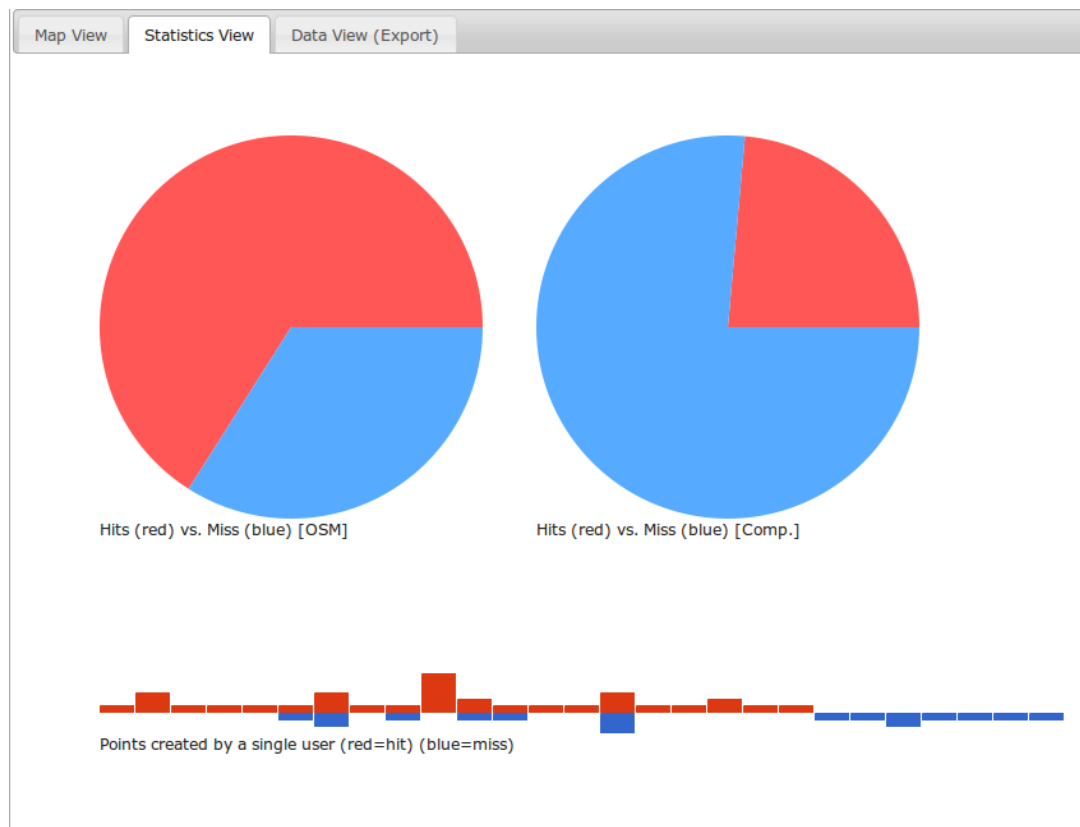


Figure 7.23: Example of the Statistics Tab

BIBLIOGRAPHY

- Jakob Altenstein. *Aufstellung und Implementierung von Kriterien zur Ermittlung des möglichen Datenverlustes im Rahmen des bei OpenStreetMap geplanten Lizenzwechsels von CC-BY-SA nach ODbL sowie Gegenüberstellung der aktuellen und der prognostizierten Datenbestände in kartographischer Form*. PhD thesis, Hochschule Karlsruhe, 2011.
- J. Chris Anderson, Jan Lehnardt, and Noah Slater. *CouchDB: The Definitive Guide Time to Relax*. O'Reilly Media, Inc., 1 edition, 2010. ISBN 0596155891, 9780596155896. URL <http://guide.couchdb.org/editions/1/en/index.html>.
- Apache Software Foundation. *Apache couchdb*, 2012. URL <http://couchdb.apache.org/>. [Online; accessed 16-September-2012].
- Aamer Ather. *A Quality Analysis of OpenStreetMap Data*. PhD thesis, University College London, 2009.
- Mark Alexander Bain. *How to use modular programming*, 4 2009. URL <http://suite101.com/article/how-to-use-modular-programming-a139055>. [Online; accessed 31-May-2012].
- Thomas Bayer and Dirk M. Sohn. *Rest web services*. *t3n*, 8, 2007.
- Bundeskanzleramt Österreich. 25. Bundesgesetz: Urheberrechtsgesetz-Novelle 1997 – UrhG-Nov 1997. *Bundesgesetzblatt von 1945 - 2003 - 25/1998*, 1998. URL https://www.ris.bka.gv.at/Dokument.wxe?Abfrage=BgblPdf&Dokumentnummer=1998_25_1.
- Howard Butler, Martin Daly, Allan Doyle, Sean Gillies, Tim Schaub, and Christopher Schmidt. *The geojson format specification*, 2008. URL <http://www.geojson.org/geojson-spec.html>. [Online; accessed 07-June-2012].
- Rick Cattell. *Scalable sql and nosql data stores*. *SIGMOD Rec.*, 39(4):12–27, 2011. ISSN 0163-5808. doi: 10.1145/1978915.1978919. URL <http://doi.acm.org/10.1145/1978915.1978919>.
- Andre Charland and Brian Leroux. *Mobile application development: web vs. native*. *Commun. ACM*, 54(5):49–53, May 2011. ISSN 0001-0782. doi: 10.1145/1941487.1941504. URL <http://doi.acm.org/10.1145/1941487.1941504>.
- Hainan Chen and Volker Walter. *Quality inspection and quality improvement of large spatial datasets*. In *Spatial Data Infrastructure Convergence: Building SDI Bridges to Address Global Challenges*, 2009. URL <http://www.gsdi.org/gsdiconf/gsdi11/papers/pdf/374.pdf>.
- Blazej Cipeluch, Ricky Jacob, Adam Winstanley, and Peter Mooney. *Comparison of the accuracy of openstreetmap for ireland with google maps and bing maps*. *Proceedings of the Ninth International*

Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences 2010, page 337, 2010.

Maria A. Cobb, Miyi J. Chung, Harold Foley, III, Frederick E. Petry, Kevin B. Shaw, and H. Vincent Miller. A rule-based approach for the conflation of attributed vector data. *Geoinformatica*, 2(1):7–35, March 1998. ISSN 1384-6175. doi: 10.1023/A:1009788905049. URL <http://dx.doi.org/10.1023/A:1009788905049>.

R. Congalton and K. Green. *Assessing the accuracy of remotely sensed data: principles and practices*. Lewis, Boca Raton, FL, 1999.

Creative Commons. Creative Commons Attribution-ShareAlike 2.0, a. URL <http://creativecommons.org/licenses/by-sa/2.0/de/legalcode>. [Online; accessed 10-June-2012].

Creative Commons. Summary of Creative Commons Attribution-ShareAlike 2.0, b. URL <http://creativecommons.org/licenses/by-sa/2.0/>. [Online; accessed 11-June-2012].

Douglas Crockford. The application/json media type for javascriptobject notation (json), 2006. URL <http://tools.ietf.org/html/rfc4627>. [Online; accessed 28-May-2012].

Daniel P. Delorey, Charles D. Knutson, and C. Giraud-Carrier. Programming language trends in open source development: An evaluation using data from all production phase sourceforge projects. In *Second International Workshop on Public Data about Software Development (WoODaSD '07)*, 2007.

Alexis Deveria. When can I use ..., 2012. URL <http://caniuse.com/#search=WebWorker>. [Online; accessed 13-June-2012].

emka. Osmcouch, 2012. URL <https://github.com/emka/OSMCouch>. [Online; accessed 16-September-2012].

European Union. Directive 96/9/ec of the european parliament and of the council of 11 march 1996 on the legal protection of databases. *Official Journal of the European Union*, 27/03(L 077):20 – 28, 1996. URL <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31996L0009:EN:HTML>.

Apache Software Foundation. Apache software foundation, 2012. URL <http://www.apache.org/foundation/>. [Online; accessed 27-May-2012].

GDAL Development Team. *GDAL - Geospatial Data Abstraction Library*. Open Source Geospatial Foundation, 2012a. URL <http://www.gdal.org>. [Online; accessed 16-September-2012].

GDAL Development Team. *CouchDB driver - Geospatial Data Abstraction Library*. Open Source Geospatial Foundation, 2012b. URL http://www.gdal.org/ogr/drv_couchdb.html. [Online; accessed 16-September-2012].

Geofabrik GmbH. Download openstreetmap extracts, 2012. URL <http://download-int.geofabrik.de/osm/europe/>. [Online; accessed 28-August-2012].

Seth Gilbert and Nancy Lynch. Brewer’s conjecture and the feasibility of consistent, available, partition-tolerant web services. *SIGACT News*, 33(2):51–59, June 2002. ISSN 0163-5700. doi: 10.1145/564585.564601. URL <http://doi.acm.org/10.1145/564585.564601>.

Michael F. Goodchild and Gary J. Hunter. A simple positional accuracy measure for linear features. *International Journal of Geographical Information Science*, 11(3):299–306, 1997. doi: 10.1080/136588197242419. URL <http://www.tandfonline.com/doi/abs/10.1080/136588197242419>.

- Google Incorporated. Google scholar, 2012. URL <http://scholar.google.at/>. [Online; accessed 28-March-2012].
- GoogleCode. ie-web-worker, 2009. URL <http://code.google.com/p/ie-web-worker/>. [Online; accessed 20-September-2012].
- Heinz-Peter Gumm and Manfred Sommer. *Einführung in die Informatik*. Oldenbourg Wissenschaftsverlag GmbH, 5 edition, 2002.
- Manish Gupta, Rui Li, Zhijun Yin, and Jiawei Han. Survey on social tagging techniques. *SIGKDD Explor. Newsl.*, 12(1):58–72, 2010. ISSN 1931-0145. doi: 10.1145/1882471.1882480. URL <http://doi.acm.org/10.1145/1882471.1882480>.
- Muki Haklay. How good is volunteered geographical information? a comparative study of open-streetmap and ordnance survey datasets. *Environment and Planning B: Planning and Design*, 37(4):682–703, 2010. URL <http://www.envplan.com/abstract.cgi?id=b35097>.
- Muki Haklay. Observing from afar or joining the action: Osm and giscience research. *State of the Map EU*, 2011. URL <http://www.sotm-eu.org/talk?41>.
- Marijn Haverbeke. *Eloquent JavaScript*. William Pollock, 1 edition, 2011.
- Marco Helbich, Christof Amelunxen, and Pascal Neis. Comparative spatial analysis of positional accuracy of openstreetmap and proprietary geodata. In *International GI Forum 2012*, Salzburg, 2012.
- Ian Hickson, Google, Inc. Web workers, 07 2012. URL <http://dev.w3.org/html5/workers/>. [Online; accessed 04-August-2012].
- Joyent Incorporated. Node.js, 2012. URL <http://nodejs.org/>. [Online; accessed 22-May-2012].
- jQuery Foundation. jQuery Projects, 2012. URL <http://jquery.org/>. [Online; accessed 24-Juli-2012].
- Neal Leavitt. Will nosql databases live up to their promise? *Computer*, 43(2):12 – 14, 2010.
- Ina Ludwig. Abbildung von Straßendaten für Qualitätsuntersuchungen - Ein Vergleich von OpenStreetMap mit Navteq. Thesis, Fraunhofer IAIS, 2010. URL <http://publica.fraunhofer.de/starweb/servlet.starweb?path=urn.web&search=urn:nbn:de:0011-n-1174865>.
- MaZderMind. osm-history-splitter, 2012. URL <https://github.com/MaZderMind/osm-history-splitter>. [Online; accessed 22-May-2012].
- McDonald's. Our company, 2012. URL http://www.aboutmcdonalds.com/mcd/our_company.html. [Online; accessed 13-October-2012].
- David Sawyer McFarland and Stefan Fröhlich. *JavaScript: Missing Manual*. O'Reilly Media, 1 edition, 2009.
- Mario Miler, Damir Medak, and Drazen Odobasic. Two-tier architecture for web mapping with nosql data- base couchdb. *Geospatial CrossroadsGIForum 11*, pages 62–71, 2011. URL http://bib.irb.hr/datoteka/519442.Two-Tier_Architecture_for_Web_Mapping_with_NoSQL_Database_CouchDB.pdf.
- Volker Mische. Creating a new indexer. Online Tutorial, 2010. URL <http://vmx.cx/couchdb/tutorial/indexer.html>.
- Volker Mische. Geocouch - geometry search branch, 2012. URL <https://github.com/vmx/geocouch/tree/couchdbgeom>. [Online; accessed 16-September-2012].

- Volker Mische, Pierre Giraud, and Filipe Manana. Geocouch - readme, 2012. URL <https://github.com/couchbase/geocouch/blob/master/README.md>. [Online; accessed 29-May-2012].
- San Murugesan, Gustavo Rossi, Linda Wilbanks, and Reza Djavanshir. The future of web apps. *IT Professional*, 13:12–14, 2011. ISSN 1520-9202. doi: <http://doi.ieeecomputersociety.org/10.1109/MITP.2011.89>.
- Martin Oczipka. *Objektbasierte Klassifizierung hochauflösender Daten in urbanen Räumen unter besonderer Berücksichtigung von Oberflächenmodellen*. PhD thesis, Freie Universität Berlin, 2007. URL <http://d-nb.info/988471426/34>.
- Roland Olbricht. Overpass API. In *FOSSGIS 2012 - Anwenderkonferenz für Freie und Open Source Software für Geoinformationssysteme*, 2012.
- Open Data Commons. Open database license (odbl), 2012a. URL <http://opendatacommons.org/licenses/odbl/1.0/>. [Online; accessed 28-March-2012].
- Open Data Commons. Database contents license (dbcl), 2012b. URL <http://opendatacommons.org/licenses/dbcl/1-0/>. [Online; accessed 18-July-2012].
- Open Data Commons. Open database license (odbl) summary, 2012c. URL <http://opendatacommons.org/licenses/odbl/summary/>. [Online; accessed 23-Juli-2012].
- Open Street Map Foundation. License, 2011a. URL <http://www.osmfoundation.org/wiki/License>. [Online; accessed 10-June-2012].
- Open Street Map Foundation. Contributor terms, 2011b. URL http://www.osmfoundation.org/wiki/License/Contributor_Terms. [Online; accessed 10-June-2012].
- Open Street Map Foundation. About, 2011c. URL <http://www.osmfoundation.org/index.php?title=OSMF:About&oldid=576>. [Online; accessed 18-July-2012].
- OpenLayers Contributors. Openlayers, 2012. URL <http://openlayers.org/>. [Online; accessed 29-August-2012].
- OpenStreetMap Wiki. Common licence interpretations, 2010. URL http://wiki.openstreetmap.org/w/index.php?title=Common_licence_interpretations. [Online; accessed 11-June-2012].
- OpenStreetMap Wiki. Proposed features, 2011. URL http://wiki.openstreetmap.org/w/index.php?title=Proposed_features. [Online; accessed 22-May-2012].
- OpenStreetMap Wiki. Research, 2012a. URL <http://wiki.openstreetmap.org/w/index.php?title=Research>. [Online; accessed 9-May-2012].
- OpenStreetMap Wiki. History, 2012b. URL <http://wiki.openstreetmap.org/w/index.php?title=History>. [Online; accessed 9-May-2012].
- OpenStreetMap Wiki. About, 2012c. URL <http://wiki.openstreetmap.org/w/index.php?title=About>. [Online; accessed 9-May-2012].
- OpenStreetMap Wiki. Stats, 2012d. URL <http://wiki.openstreetmap.org/w/index.php?title=Stats>. [Online; accessed 9-May-2012].
- OpenStreetMap Wiki. Osm xml, 2012e. URL http://wiki.openstreetmap.org/w/index.php?title=OSM_XML. [Online; accessed 22-May-2012].
- OpenStreetMap Wiki. Pbf format, 2012f. URL http://wiki.openstreetmap.org/w/index.php?title=PBF_Format. [Online; accessed 22-May-2012].

- OpenStreetMap Wiki. Api v0.6, 2012g. URL http://wiki.openstreetmap.org/w/index.php?title=API_v0.6. [Online; accessed 22-May-2012].
- OpenStreetMap Wiki. Xapi, 2012h. URL <http://wiki.openstreetmap.org/w/index.php?title=Xapi>. [Online; accessed 23-May-2012].
- OpenStreetMap Wiki. Open data license/substantial - guideline, 2012i. URL http://wiki.openstreetmap.org/w/index.php?title=Open_Data_License/Substantial_-_Guideline. [Online; accessed 2-August-2012].
- OpenStreetMap Wiki. Open data license/produced work - guideline — openstreetmap wiki,, 2012j. URL http://wiki.openstreetmap.org/w/index.php?title=Open_Data_License/Produced_Work_-_Guideline. [Online; accessed 4-August-2012].
- OpenStreetMap Wiki. Key:amenity, 2012k. URL <http://wiki.openstreetmap.org/w/index.php?title=Key:amenity&oldid=781207>. [Online; accessed 28-August-2012].
- OpenStreetMap Wiki. Main page, 2012l. URL http://wiki.openstreetmap.org/w/index.php?title=Main_Page. [Online; accessed 28-March-2012].
- Kai Orend. *Analysis and Classification of NoSQL Databases and Evaluation of their Ability to Replace an Object-relational Persistence Layer*. Master's thesis, Technical University Munich, April 2010.
- John K. Ousterhout. Scripting: Higher-level programming for the 21st century. *Computer*, 31(3):23–30, March 1998. ISSN 0018-9162. doi: 10.1109/2.660187. URL <http://dx.doi.org/10.1109/2.660187>.
- Ariel Ortiz Ramirez. Three-tier architecture. *Linux Journal*, (75), 2000. URL <http://www.linuxjournal.com/article/3508>. [Online; accessed 31-May-2012].
- Frederik Ramm, Jochen Topf, and Steve Chilton. *OpenStreetMap*. UIT Cambridge, 1 edition, October 2010.
- Roland Ramthun. *Offene Geodaten durch OpenStreetMap*, pages 159 – 184. 2012.
- Joonas Salo. Pattern definition of mapreduce. In *vikingPLoP*, 2012. URL <http://vikingplop.org/papers/8.pdf>.
- Manuela Schmidt and Pascal Neis. OpenStreetMap in der Forschung? In *FOSSGIS - Freie und Open Source Software für Geoinformationssysteme*, 2011.
- Mareike Schoof. ATKIS-Basis-DLM und OpenStreetMap - Ein Datenvergleich anhand ausgewählter Gebiete in Niedersachsen. Master's thesis, Universität Osnabrück, 2010.
- Alexander Strunck. *Raumzeitliche Qualitätsuntersuchung von OpenStreetMap*. PhD thesis, Rheinischen Friedrich-Wilhelms-Universität Bonn, 2010.
- TIOBE Software BV. Tiobe programming community index for may 2012, 2012. URL <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>. [Online; accessed 15-May-2012].
- Oliver Tonnhofer. Imposm geocouch branch, 2012. URL <https://bitbucket.org/olt/imposm/src>. [Online; accessed 16-September-2012].
- Jochen Topf. Taginfo, 2012a. URL <http://taginfo.openstreetmap.org/>. [Online; accessed 22-May-2012].
- Jochen Topf. Osmjs, 2012b. URL <https://github.com/joto/osmium/tree/master/osmjs>. [Online; accessed 18-September-2012].

Tupalo Internetservices GmbH. Tupalo.com holt sich den herold, 2009. URL <http://tupalo.com/de/press/2009/06/tupalocom-holt-sich-den-herold/>. [Online; accessed 28-August-2012].

Tupalo Internetservices GmbH. Tupalo.com, 2012. URL <http://tupalo.com/de/>. [Online; accessed 23-Juli-2012].

Unity Technologies. Unity3d game engine, 2012. URL <http://unity3d.com/>. [Online; accessed 22-May-2012].

Gareth Watts. Sparklines, 2012. URL <http://omnipotent.net/jquery.sparkline>. [Online; accessed 21-September-2012].

Andreas Wenk and Till Klampäcker. *CouchDB*. Galileo Computing, Bonn, 2011.

Stadt Wien. Open government data, 2012. URL <http://data.wien.gv.at/>. [Online; accessed 20-September-2012].