Many-To-Many Polygon Matching à La Jaccard

Alexander Naumann[1]
University of Bonn, Germany

Annika Bonerath
University of Bonn, Germany

Jan-Henrik Haunert
University of Bonn, Germany

Abstract

Integration of spatial data is a major field of research. An important task of data integration is finding correspondences between entities. Here, we focus on combining building footprint data from cadastre and from volunteered geographic information, in particular OpenStreetMap. Previous research on this topic has led to exact 1:1 matching approaches and heuristic m:n matching approaches, most of which are lacking a mathematical problem definition. We introduce a model for many-to-many polygon matching based on the well-established Jaccard index. This is a natural extension to the existing 1:1 matching approaches. We show that the problem is NP-complete and a naive approach via integer programming fails easily. By analyzing the structure of the problem in detail, we can reduce the number of variables significantly. This approach yields an optimal m:n matching even for large real-world instances with appropriate running time. In particular, for the set of all building footprints of the city of Bonn (119,300 / 97,284 polygons) it yielded an optimal solution in approximately 1 hour.

2012 ACM Subject Classification Information systems → Geographic information systems; Theory of computation → Computational geometry; Theory of computation → Integer programming; Mathematics of computing → Matchings and factors

Keywords and phrases polygon matching, exact algorithm, Jaccard index

Digital Object Identifier 10.4230/LIPIcs.ESA.2024.90

Supplementary Material
Software (Source Code): `https://github.com/anaumannigg/PolyMatching`

Funding *Alexander Naumann*: Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Projektnummer 459420781.
Annika Bonerath: German Research Foundation under Germany's Excellence Strategy, EXC-2070 - 390732324 - PhenoRob.

1 Introduction

When dealing with spatial data, it is often necessary to integrate information from several data sources of the same region. Consider the scenario of building footprints available on the one hand as cadastral data, e.g., in Germany ALKIS [2], and on the other hand data created by volunteers, in our case OpenStreetMap [15]. These data sets most likely differ in the number of buildings as well as the buildings' exact positions and outlines. For example, an OpenStreetMap user registers a building complex as a whole, while the official land registry lists the complex as a conglomerate of multiple buildings. Often the buildings of both data sets have different attributes, such as type, height, zip code, year of construction, etc. Thus,

[1] corresponding author

32nd Annual European Symposium on Algorithms (ESA 2024).
Editors: Timothy Chan, Johannes Fischer, John Iacono, and Grzegorz Herman; Article No. 90; pp. 90:1–90:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

it is desirable to integrate both data sets, where a major task is finding correspondences between the buildings.

Besides data integration, finding such correspondences allows a quality assessment of one map with respect to a benchmark map. This is especially useful in the active research field of generating maps from satellite data [20].

In this paper, we discuss the problem of finding matches (correspondences) between elements in the two sets of polygons B_1 and B_2. For example, every element $p \in B_1$ is a polygon of a building footprint of the ALKIS data and every element $q \in B_2$ is a polygon of a building footprint of the OpenStreetMap data. Note that 1:1 matches do not suffice to model the relationships between building footprints in real-world data (see Figure 1 (a.)). The same holds for 1:n matches. Hence, we allow m:n (many-to-many) matches, where every match μ is defined with a subset of B_1 and a subset of B_2, i.e., $\mu = \{P_\mu \subseteq B_1, Q_\mu \subseteq B_2\}$. We call a set of matches a *many-to-many matching* (or *m:n matching*) if all polygons in B_1 and B_2 are part of at most one match.

Amongst all possible m:n matchings, we want to find an m:n matching M that consists of reasonable matches. As a widely applied measure, which has been used before to rate 1:1 matchings, we make use of the *Jaccard index* IoU (intersection over union) which for two-dimensional objects P_μ, Q_μ is defined via the area of the intersection $\mathrm{I}(P_\mu, Q_\mu)$ and the area of the union $\mathrm{U}(P_\mu, Q_\mu)$.

$$\mathrm{IoU}(\mu) = \mathrm{IoU}(P_\mu, Q_\mu) = \frac{\mathrm{I}(P_\mu, Q_\mu)}{\mathrm{U}(P_\mu, Q_\mu)} \tag{1}$$

Note, however, that not all matches with IoU > 0 might be favorable: Figure 1 (b.) shows an example, where two intersecting building footprints of the two data sets clearly do not correspond to each other. To avoid these matches, we introduce a parameter $\lambda \in [0, 1)$ to quantify the quality of a match μ as $\sigma(\mu) = \mathrm{IoU}(\mu) - \lambda$.

Overall, we want to find an *optimal* m:n matching, i.e., an m:n matching maximizing the sum over the qualities of the contained matches.

$$\max_{M \text{is } m:n \text{ matching}} \sum_{\mu \in M} (\mathrm{IoU}(\mu) - \lambda) \tag{2}$$

We call $\sigma(M) = \sum_{\mu \in M} (\mathrm{IoU}(\mu) - \lambda)$ the quality of an m:n matching M.

▶ **Problem 1** (Matching).

Given: *Data sets B_1 and B_2 of interior-disjoint polygons.*

Find: *Many-to-many matching M^* that maximizes the quality $\sigma = \sum_{\mu \in M^*} (\mathrm{IoU}(\mu) - \lambda)$.*

To the best of our knowledge, this is the first formal definition of many-to-many polygon matching, while previous research focused on heuristic approaches to the problem.

We prove by reduction from `PARTITION` [9] that the decision variant of our optimization problem is NP-complete (Subsection 3.1). This motivates the use of integer linear programming (ILP). However, trying to solve the problem with a naive ILP formulation easily gets infeasible. We present structural properties of the problem that allow us to solve even large data sets optimally. For example, we show that our choice of objective implies that the polygons of a match included in an optimal m:n matching are connected (Section 4). Using structural properties, we can decompose the instance into smaller sub-problems (Section 5). We implemented the algorithm and applied it to large-scale real-world data comparing ALKIS and OpenStreetMap data from Bonn and Cologne, Germany, to confirm its applicability with experiments (Section 6).

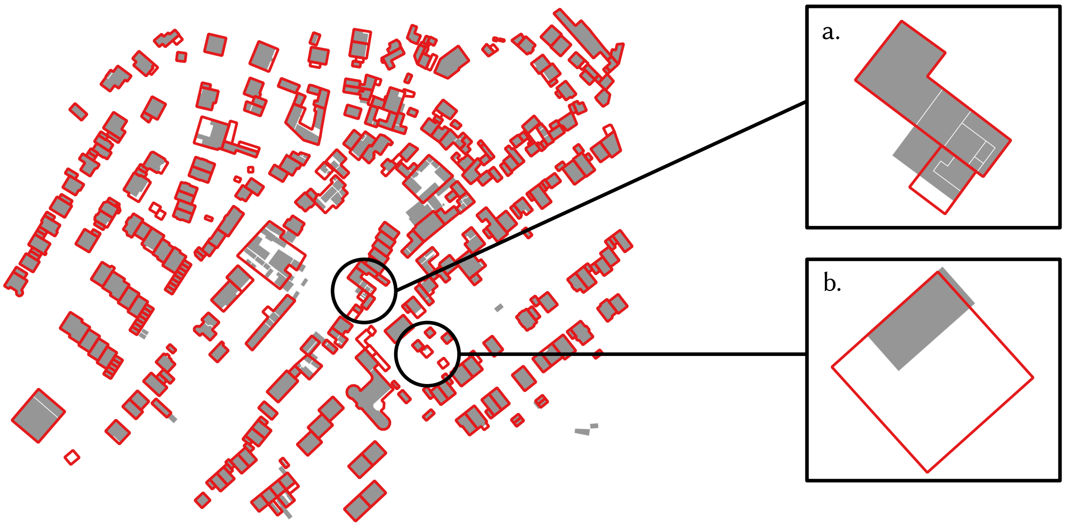

Figure 1 An excerpt of the data sets from the city of Bonn (ALKIS in grey, OSM in red). ((a.) An example occurring in the excerpt where a 1:1 match does not suffice. (b.) Two footprints of the excerpt that do intersect but should not be matched.)

Our Contribution

With this work, we investigate correspondences between two sets of interior-disjoint polygons. In particular, we contribute

- a model for an optimal many-to-many matching,
- an NP-completeness proof of the problem,
- an analysis of structural properties,
- an ILP formulation for solving the problem,
- an algorithm using the structural properties to reduce the number of variables and constraints of the ILP,
- experiments on large-scale real-world data.

2 Related Work

For the matching of polygons representing building footprints, multiple approaches have been introduced. Fan et al. proposed a heuristic approach for finding m:n matchings of building footprints [5]. Two buildings are in the same match if their intersection area divided by the area of the smaller polygon is larger than a threshold. In application, often simpler matching rules are used. For example, Müller et al.'s research focuses on quality assessment [12]. The matching is a preprocessing step where they match every polygon from one data set to the polygon from the other data set that has the smallest centroid distance. While both works provide an evaluation of real-world data, they do not introduce a formal statement of the problem. In contrast, our approach is based on a formally stated optimization problem with a clear optimization objective.

Liu et al. present an approach *MBRCO* for matching polygons of two maps of different scales [11]. In contrast to our problem, corresponding objects can be dislocated due to shape simplification. Their approach is based on the minimum bounding rectangles of the input polygons where the quality of a match is rated by the consistency of the shapes. However, they neither state the problem formally nor provide proofs regarding the optimality.

Besides approaches that are based on geometry (location and shape), there also exists work that incorporates other information. For example, Kim et al. use geographical context to find the matching [10]. Huh et al. consider the problem of finding a matching of automatically retrieved building footprints from satellite image data [24]. They use latent semantic analysis where they assign a feature vector to each polygon and perform a hierarchical clustering.

Closely related to matchings of polygons, there exists work on matchings of lines that represent a road network. Walter et al. introduced a method that grows buffers around entities [21]. Based on this method different approaches have been developed, e.g., taking into account statistic measures [22] or extending the lines additionally [23]. Again, these methods are heuristic approaches.

For comparing two maps, evaluating found correspondences of entities is essential. Jozdani and Chen investigated several quality measures comparing two maps [8]. They showed that the Jaccard index is well-performing and robust. Also, the Jaccard index has been widely applied in related problems, e.g., in the context of real-time tracking with 1:1 matches [1] and m:n matches [13], and for the quality assessment of extracted polygons from satellite imagery to cadastral data [20]. Hecht et al. proposed an approach that quantifies a map's completeness by measuring the overlap with respect to a benchmark map [7]. Müller et al. introduced the difference between integrals of the turning function of the polygons for measuring the quality of a map [12]. Another common measure is to examine the point-to-point relationship of the matched geometries [5, 17]. We deem that the Jaccard index is one of the most common measures that is widely accepted. Hence, we use it for our objective.

Within our algorithm, we model the configuration of the two data sets as a bipartite graph. The emerging problem is known as many-to-many (graph) matching. Regarding this problem, there has been research presenting a continuous relaxation approach [25]. From the field of computer vision, it has been explored for object [18] and pattern recognition [4]. Our problem differs from these approaches, since the qualities of our m:n matches cannot be derived directly from single buildings contained in them. Further research has also added constraints to the problem in order to find efficient algorithms. Examples are critical matchings introducing lower quotas to each vertex [14] or tree-constrained matchings considering two trees as parts of the bipartite graph, where the tree hierarchies define the possible sets to be matched [3].

3 Model

In the following, we consider the presented problem as a graph problem. For this, let $G = (V = \{V_1 \cup V_2\}, E)$ be a graph where for $i \in \{1, 2\}$ a vertex in V_i represents a set of polygons from B_i and an edge $e \in V_1 \times V_2$ corresponds to a possible match $\mu = (P \subseteq B_1, Q \subseteq B_2)$. Each edge $e_\mu \in E$ is weighted with the match quality $w(e_\mu) = \sigma(\mu)$. We denote the set of represented polygons of a vertex v of G by $p(v)$. Analogously, $p(V)$ denotes the set of represented polygons of a set V of vertices. We call a vertex v *simple* if $p(v)$ has size 1 and *cumulative* otherwise.

An m:n matching M corresponds to a constrained matching in G such that no polygon occurs in more than one set of polygons represented by a vertex incident to an edge of the matching. We call such a selection of edges *matching selection*. An optimal m:n matching corresponds to the matching selection with the highest sum of edge weights over all matching selections. We call a graph G containing the optimal matching selection a *candidate graph*. A trivial example of a candidate graph is a complete bipartite graph where V_i contains a vertex for every element of the powerset of B_i, see Figure 2.

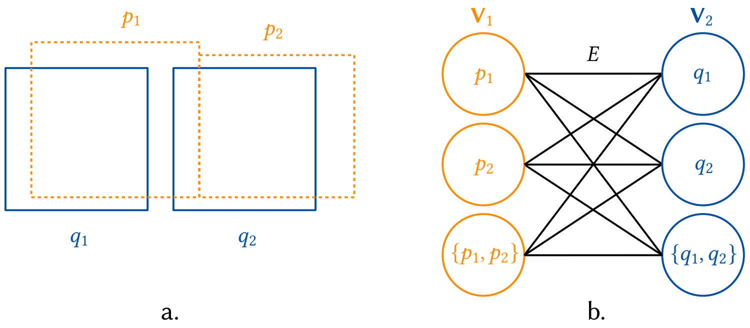

Figure 2 An example of (a.) two sets of polygons (blue and orange) and (b.) the candidate graph G we use to compute the optimal $m{:}n$ matching.

The presented $m{:}n$ matching problem can easily be formulated as an integer linear program (ILP). For each edge $e \in E$, we introduce a variable $x_e \in \{0, 1\}$. To ensure that a polygon p is contained in at most one match, we introduce the constraint $\sum_{e \in E_p} x_e \leq 1$ where E_p is the set of edges incident to any vertex v where $p \in p(v)$. Formally, this gives us the following objective for the ILP.

$$\textbf{maximize} \sum_{e \in E} w(e) \cdot x_e \tag{3}$$

3.1 NP-Completeness

▶ **Theorem 1.** *The problem of deciding for a given $\sigma^* \in \mathbb{R}$ whether an $m{:}n$ matching M^* with quality $\sigma(M^*) > \sigma^*$ exists is NP-complete.*

Proof. Let MATCHING be the problem of deciding whether for a given quality $\sigma^* \in \mathbb{R}$ there exists an $m{:}n$ matching M^* with $\sigma(M^*) \geq \sigma^*$. To prove NP-completeness of MATCHING, we need to show that it lies in NP: Given an $m{:}n$ matching M, the matching quality $\sigma(M)$ can be computed in polynomial time. Hence, deciding whether $\sigma(M) \geq \sigma^*$ can be done in polynomial time and MATCHING $\in$ NP.

We prove NP-completeness of MATCHING via a polynomial reduction PARTITION $\subseteq_p$ MATCHING (PARTITION is NP-complete [9]). Let $A = \{a_1, ..., a_n\}$ be a set where for all $i \in [n]$: $a_i \in \mathbb{Z}$. Then, PARTITION decides if there is a subset $A' \subset A$ such that $\sum_{a \in A'} a = \sum_{a \in (A \setminus A')} a$. We will build an instance of MATCHING from the given instance of PARTITION as depicted in Figure 3. The optimal solution of the constructed MATCHING instance will enable us to solve the underlying PARTITION problem.

Let $p_1, \ldots, p_n$ be axis-parallel rectangular polygons in $\mathbb{R}^2$ such that the upper left corner of rectangle p_i is $(\sum_{j=1}^{i-1} a_j, 2)$ and the lower right corner is $(\sum_{j=1}^{i} a_j, 0)$. Let $S = \sum_{a \in A} a$. Let p_{n+1} be a polygon defined by $(S, 2) \times (2S, 1)$ and p_{n+2} be a polygon defined by $(S, 1) \times (2S, 0)$. Let B_1 be the set $\{p_1, \ldots, p_{n+2}\}$. Let B_2 be a set of two rectangular polygons q_1, q_2 defined by $(0, 2) \times (2S, 1)$ and $(0, 1) \times (2S, 0)$. These polygons are the input of MATCHING. We can construct them in polynomial time. We set $\lambda = 0$.

Now, we will show that $\sigma^* = \frac{6}{5}$ is the optimal quality for the given instance and that it can be achieved only for a $m{:}n$ matching $\{\{B_1', q_1\}, \{B_1 \setminus B_1', q_2\}\}$ where B_1' corresponds to a set $A' \subset A$ that fulfills PARTITION. Due to $|B_2| = 2$, we have two options to match the polygons from B_2. The first option is that both polygons of B_2 are in the same match. Then, trivially, it is optimal to match them to all polygons of P leading to a quality $\sigma(\{B_1, B_2\}) = 1$. We call this $m{:}n$ matching $M_{B_{12}}$. The second option is to separate the polygons of B_2 into two

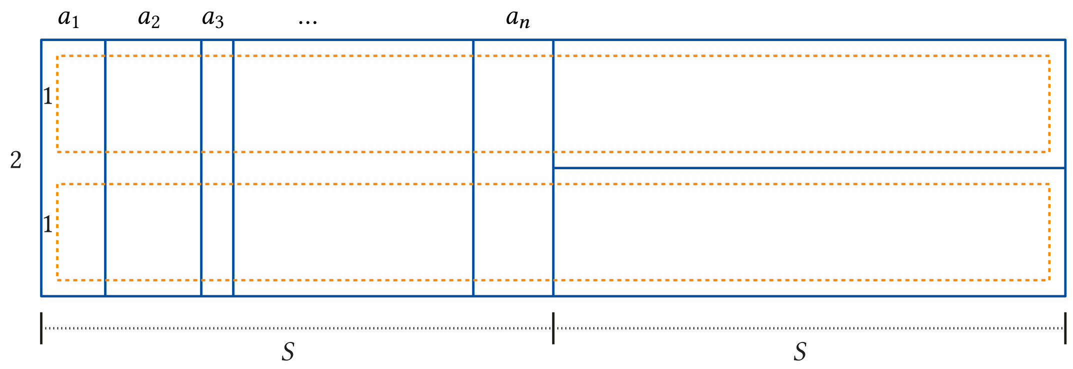

Figure 3 Construction of an instance of the `MATCHING` from an instance of `PARTITION`. The blue polygons correspond to polygons of B_1 and the orange polygons to polygons of B_2. For the sake of readability, we scaled down the size of the polygons in B_2.

matches. Let $\mu_1 = \{B_1', q_1\}$ and $\mu_2 = \{B_1 \setminus B_1', q_2\}$ and $M = \{\mu_1, \mu_2\}$, where B_1' corresponds to a set A' with an element wise sum of exactly $S/2$. For M, we obtain

$$\sigma(M) = \mathrm{IoU}(\mu_1) + \mathrm{IoU}(\mu_2) = \frac{1.5S}{2.5S} + \frac{1.5S}{2.5S} = \frac{6}{5} > 1 = \sigma(M_{B_{12}}).$$

Hence, the quality of M is better than the quality of $M_{B_{12}}$. It remains to show that there is no $\hat{B}_1 \subseteq B_1$ such that $\sigma(\{\{\hat{B}_1, q_1\}, \{B_1 \setminus \hat{B}_1, q_2\}\}) > \frac{6}{5}$. Since we know that q_1, q_2 cannot be in the same match and improve upon $\sigma(M) = \frac{6}{5}$, every other possible $m{:}n$ matching can be modeled as reassigning a set of polygons $R_1 \subseteq B_1'$ from μ_1 to μ_2 and a set of polygons $R_2 \subseteq B_1 \setminus B_1'$ from μ_2 to μ_1. Let x_1 be the sum of all widths of polygons from R_1, x_2 be the sum of all widths of polygons from R_2, and $x = |x_1 - x_2|$. Note that $x \in (0, S/2]$ as the polygons of B_1' have a width of $S/2$. The quality of the new $m{:}n$ matching after reassignment corresponds to $f\colon x \mapsto \frac{1.5S+x}{2.5S+x} + \frac{1.5S-x}{2.5S-x}$ for $x \in [0, S/2]$. The maximal value of f corresponds to the maximal value of the quality over all $m{:}n$ matchings. We derive f as $f'(x) = -\frac{160S^2x}{(2x-5S)^2(2x+5S)^2}$ and $f''(x) = \frac{160S^2(12x^2+25S^2)}{(2x-5S)^3(2x+5S)^3}$ and obtain $f'(x) = 0$ only for $x = 0$ and $f''(0) < 0$. Hence, $x = 0$ is the only maximum of f, which means that assigning exactly half of the width to each match is optimal.

Hence, we can solve `PARTITION` by solving `MATCHING`: If and only if there is an $m{:}n$ matching M with a quality $\sigma(m) \geq \sigma^* = \frac{6}{5}$, then there is a solution for `PARTITION`. Hence, `MATCHING` is NP-complete. ◀

4 Structural Properties

In this section, we discuss properties of an optimal solution M^*, which we will use for our algorithm. Let $G_{\text{int}} = (V_{\text{int}} = (V_{\text{int},1} \cup V_{\text{int},2}), E_{\text{int}})$ be a bipartite graph that contains a vertex in $V_{\text{int},1}$ for every polygon in B_1 and a vertex in $V_{\text{int},2}$ for every polygon in B_2 and an edge for every two vertices where the associated polygons intersect. We call G_{int} the *intersection graph* of B_1 and B_2. We call a match $\mu = \{P \subseteq B_1, Q \subseteq B_2\}$ *connected* if the corresponding vertices induce a connected subgraph in G_{int}.

▶ **Lemma 2** (Match Connectedness). *Given two sets of polygons B_1, B_2, every matching M^* maximizing $\sigma(M^*)$ exclusively consists of connected matches.*

Proof. Let $\mu = \{P, Q\}$ be a non-connected match. Assume, there was an optimal solution including μ. We can split it into two disjoint non-empty matches $\mu_1 = \{P_1, Q_1\}$ and $\mu_2 = \{P_2, Q_2\}$ where $(P_1 \cup Q_1) \cap (P_2 \cup Q_2) = \varnothing$. We distinguish two cases:

Case 1: $\sigma(\mu_1) = \sigma(\mu_2)$. In this case, it trivially holds that $\sigma(\mu) = \sigma(\mu_1) = \sigma(\mu_2)$ and $\sigma(\mu) > 0$, since μ is in an optimal solution per assumption. But then, $\sigma(\mu) < 2 \cdot \sigma(\mu) = \sigma(\mu_1) + \sigma(\mu_2)$ and therefore the many-to-many matching could be improved by replacing μ with μ_1 and μ_2.

Case 2: w.l.o.g. $\sigma(\mu_1) < \sigma(\mu_2)$. Let I, U be the intersection and union of μ and I_i, U_i be the intersection and union of μ_i for $i \in \{1, 2\}$. We claim that $\mathrm{IoU}(\mu) < \mathrm{IoU}(\mu_2)$, which implies $\sigma(\mu) < \sigma(\mu_2)$ and therefore replacing μ by μ_2 increases the objective. This holds, because:

$$\begin{aligned}
&\mathrm{IoU}(\mu) < \mathrm{IoU}(\mu_2) && (4)\\
\Leftrightarrow\ & \frac{I}{U} < \frac{I_2}{U_2} && (5)\\
\Leftrightarrow\ & \frac{I_1 + I_2}{U_1 + U_2} < \frac{I_2}{U_2} && (6)\\
\Leftrightarrow\ & I_1U_2 + I_2U_2 < I_2U_1 + I_2U_2 && (7)\\
\Leftrightarrow\ & \frac{I_1}{U_1} < \frac{I_2}{U_2} && (8)
\end{aligned}$$

where the last equation is true by assumption since $\sigma(\mu_1) < \sigma(\mu_2)$. Hence the objective of any solution including a non-connected match μ can always be improved, meaning no many-to-many matching including such a non-connected match is optimal. ◀

▶ **Lemma 3** (Outlier Polygons). *Given two sets of polygons B_1, B_2. Let $p \in B_1$ and let Q be the subset of B_2 that consists of all polygons that intersect p. If $\frac{\mathrm{I}(p,Q)}{\mathrm{area}(p \setminus Q)} < \lambda$, then p is not part of a match in an optimal m:n matching.*

Proof. Assume p is a polygon that fulfills the property of the lemma and assume it is part of a match μ_p in an optimal solution M^*. Then, it holds that $\sigma(\mu_p) \geq 0$ and hence, $\mathrm{IoU}(\mu_p) \geq \lambda$. Assume we remove p from its match μ_p. Let x be the intersection area and y be the union area that μ_p loses when removing p. Since $\frac{\mathrm{I}(p,Q)}{\mathrm{area}(p \setminus Q)} < \lambda$, we know that $\frac{x}{y} < \lambda$. We claim that if we remove p from the match μ_p, this increases its IoU:

$$\begin{aligned}
\mathrm{IoU}(\mu_p) = \frac{\mathrm{I}(\mu_p)}{\mathrm{U}(\mu_p)} &< \frac{\mathrm{I}(\mu_p) - x}{\mathrm{U}(\mu_p) - y} = \mathrm{IoU}(\mu_p \setminus p) && (9)\\
\Leftrightarrow \mathrm{I}(\mu_p) \cdot (\mathrm{U}(\mu_p) - y) &< \mathrm{U}(\mu_p) \cdot (\mathrm{I}(\mu_p) - x) && (10)\\
\Leftrightarrow \mathrm{I}(\mu_p) \cdot y &> \mathrm{U}(\mu_p) \cdot x && (11)\\
\Leftrightarrow \frac{\mathrm{I}(\mu_p)}{\mathrm{U}(\mu_p)} &> \frac{x}{y} && (12)
\end{aligned}$$

Equation (12) always holds, since $\mathrm{IoU}(\mu_p) \geq \lambda$ and $\frac{x}{y} < \lambda$. But then, we can remove p from μ_p and increase the objective of M^*. Therefore M^* cannot have been optimal in the first place, contradicting our assumption. ◀

▶ **Lemma 4** (Polygons in the same Match). *Let $p \in B_1$, $q \in B_2$, and let $N_p(q) = \{p' \in B_1 \setminus p \mid p' \cap q \neq \emptyset\}$. If it holds that*

$$\textit{(1)}\ \mathrm{I}(p, q) > \mathrm{area}(p \setminus q), \qquad \textit{(2)}\ \mathrm{IoU}(p, q) > \lambda, \textit{ and } \qquad \textit{(3)}\ \frac{\mathrm{I}(q, N_p(q))}{\mathrm{area}(q \setminus N_p(q))} < \lambda,$$

then, polygons p and q are contained in the same match in every optimal solution.

Proof. We show this lemma via contradiction. Consider M is an optimal m:n matching, and p and q fulfill properties (1), (2), and (3), and p and q are not in the same match in M. We distinguish the following cases:

Case 1: p and q are both unmatched. Let $\mu = \{p, q\}$. Due to (2) adding μ to M will increase $q(M)$. Therefore M cannot be optimal.

Case 2: one polygon is in a match, the other one not. Assume p is in a match μ_p but q is unmatched. Then, $\mathrm{IoU}(\mu_p) > \lambda$. Let μ_{pq} be the match μ_p extended by q. Due to property (1), adding q to μ_p increases the intersection of the match more than its union. In detail, $\mathrm{I}(\mu_p \cup q) - \mathrm{I}(\mu_p) > \mathrm{U}(\mu_p \cup q) - \mathrm{U}(\mu_p)$. Note that (i) $\mathrm{IoU} \leq 1$ and (ii) $\forall a, b, c, d \in \mathbb{R}_{>0}$ with $a \leq b, c \leq d$ it holds that $\frac{a}{b} < \frac{a+d}{b+c}$. Altogether, it holds that

$$\mathrm{IoU}(\mu_p) = \frac{\mathrm{I}(\mu_p)}{\mathrm{U}(\mu_p)} < \frac{\mathrm{I}(\mu_p \cup q)}{\mathrm{U}(\mu_p \cup q)} = \mathrm{IoU}(\mu_{pq}). \tag{13}$$

Hence, M can not be optimal. Note that the case q is in a match and p is unmatched can be handled analogously.

Case 3: p and q are in different matches. Let p be in μ_p and q in μ_q with $\mu_p \neq \mu_q$. Due to property (3) and $\mathrm{IoU}(\mu_p) > \lambda$ and $\mathrm{IoU}(\mu_q) > \lambda$, we can remove p from μ_p and q from μ_q and improve the matches' contribution to the objective function.

$$\lambda < \mathrm{IoU}(\mu_p) = \frac{\mathrm{I}(\mu_p)}{\mathrm{U}(\mu_p)} < \frac{\mathrm{I}(\mu_p) - \mathrm{I}(p, \mu_p)}{\mathrm{U}(\mu_p) - \mathrm{area}(p \setminus \mu_p)} = \mathrm{IoU}(\mu_p \setminus p) \tag{14}$$

The same holds for μ_q analogously. Additionally, due to (2), $\sigma(\{p, q\}) > 0$ and adding $\{p, q\}$ to M improves the quality. Hence, M can not be optimal. ◀

We provide the following lemmas without proofs. Lemma 5 follows immediately from Lemma 4. Lemma 6 requires a contradictory argument similar to the proof of Lemma 3. Lemma 7 follows from the foregoing results.

▶ **Lemma 5** (Containment). *Given two sets of polygons B_1 and B_2. Let $p \in B_1$. If there exists $q \in B_2$ such that q contains p ($p \subset q$), then in every optimal m:n matching p is either in the same match as q or in no match at all.*

▶ **Lemma 6** (Two Matches better than one). *Given two sets of polygons B_1, B_2. Let $p, p' \in B_1$ and let $q, q' \in B_2$ with $p \neq p'$ and $q \neq q'$. If $\sigma(\{p, q\}) + \sigma(\{p', q'\}) > 1 - \lambda$, then there is no optimal m:n matching with a match that includes p, p', q, and q'.*

▶ **Lemma 7** (1:1 Match). *Given two sets of polygons B_1, B_2. Let $p \in B_1$ and $q \in B_2$ with $\sigma(\{p, q\}) > 0$. Let $P = \{p' \in B_1 \mid q \cap p' \neq \emptyset\}$ and $Q = \{q' \in B_2 \mid p \cap q' \neq \emptyset\}$. If*

1. *for every $p' \in P$ there exists $q' \in B_2 \setminus q$ such that $q(\{p, q\}) + q(\{p', q'\}) > 1 - \lambda$*
2. *for every $q' \in Q$ there exists $p' \in B_1 \setminus p$ such that $q(\{p, q\}) + q(\{p', q'\}) > 1 - \lambda$*
3. *Lemma 4 (polygons in the same match) holds for (p', q'),*

then every optimal solution contains the 1:1 match $\{p, q\}$.

5 Construction of Candidate Graph

Our algorithm to compute an optimal m:n matching (Algorithm 1) consists of two parts. First, we build the candidate graph and, secondly, we use an ILP to solve the problem. In the following, we will describe our procedure of setting up the candidate graph, as the ILP is provided in Section 3.

Algorithm 1 OPTIMAL M:N MATCHING.

```
    Input polygon-sets $\{B_1, B_2\}$, threshold $\lambda$
    Output optimal $m$:$n$ matching $M^*$
 1: $M^* \leftarrow \{\}, G \leftarrow \{\}$
 2: compute intersection graph $G_{\text{int}} = (V_{\text{int}}, E_{\text{int}})$ of $B_1, B_2$ and initialize $G = G_{\text{int}}$
 3: $\mathcal{C} \leftarrow$ find all connected components of $G$
 4: for each connected component $C \in \mathcal{C}$ do
 5:     insert one representing vertex for each included group to $G$ (Lemma 4)
 6:     compute 1:1 matches and delete the respective vertices from $G$ (Lemma 7)
 7:     add cumulative_vertices($C$) to $G$
 8: set up Integer Linear Program using $G$
 9: $M^* \leftarrow$ solve Integer Linear Program
10: return $M^*$
```

Let B_1 and B_2 be the input sets of polygons and G a candidate graph. To improve upon the naive approach of building a graph with every combinatorial subset of B_1 and B_2, we make use of the structural properties of any optimal solution (Section 4).

We start by computing the intersection graph $G_{\text{int}} = (V_{\text{int}} = (V_{\text{int},1}, V_{\text{int},2}), E_{\text{int}})$ of B_1 and B_2 and initialize the candidate graph with $G = G_{\text{int}}$. We decompose G into its connected components $\mathcal{C}$. Due to Lemma 2 (Match Connectedness), we can handle those as independent sub-instances. For each connected component we perform three steps: First, we use Lemma 4 (Polygons in the same Match), which means we search for sets of vertices V with a common neighbor v', where the represented polygons $p(V)$ and $p(v')$ fulfill properties (1), (2), and (3). Then the polygons in $p(V)$ are always in the same match. Hence, we replace V by one single vertex $\tilde{v}$ in G representing all polygons $p(V)$ (line 5). Secondly, we apply Lemma 7. If we can find a pair of vertices v, v' where $p(v)$ and $p(v')$ fulfill properties 1., 2., 3. of Lemma 7 (1:1 Match) we add the match $\{p(v), p(v')\}$ to our optimal m:n matching M^* and we remove v and v' from G (line 6). Thirdly, we build the cumulative vertices and their edges and add these to G' (line 7). For that, we traverse G in a breadth-first-way from each of its vertices once. In contrast to the standard breadth-first-search, our queue holds sets of vertices S, where each set corresponds to a possible match $\mu_S = \{S \cap V_1, S \cap V_2\}$. In every iteration, we pop such a set S off the queue. We add the corresponding cumulative vertices to G if they are not included yet. A weighted edge between the vertices is added if the quality of the corresponding match is larger than 0. For each vertex v in G adjacent to a vertex within S, we insert the set $S \cup \{v\}$ into the queue if it has not been visited yet. This way, we find every set of vertices that could potentially form a match in M^* and G is a candidate graph afterward. We speed up our computation via the following techniques:

- Note that sets can be found more than once. Hence, an efficient way of checking for duplicates is necessary. We make use of an AVL tree data structure.
- To avoid visiting a large number of matches that can never be in an optimal solution we can stop the breadth-first search if, for the polygons $p(S)$ of the current candidate match, there is a 1:1 matching $M_{1:1}$ of $p(S)$ with $\sigma(M_{1:1}) > 1 - \lambda$. We use a simple greedy heuristic to check this.
- To compute the weights of the edges in the candidate graph, we need many redundant spatial computations (intersection, union). We reduce the running time by pre-computing an arrangement of all line segments of the polygons in both sets. This way we can compute the area per face once and only need to make look-ups later on.

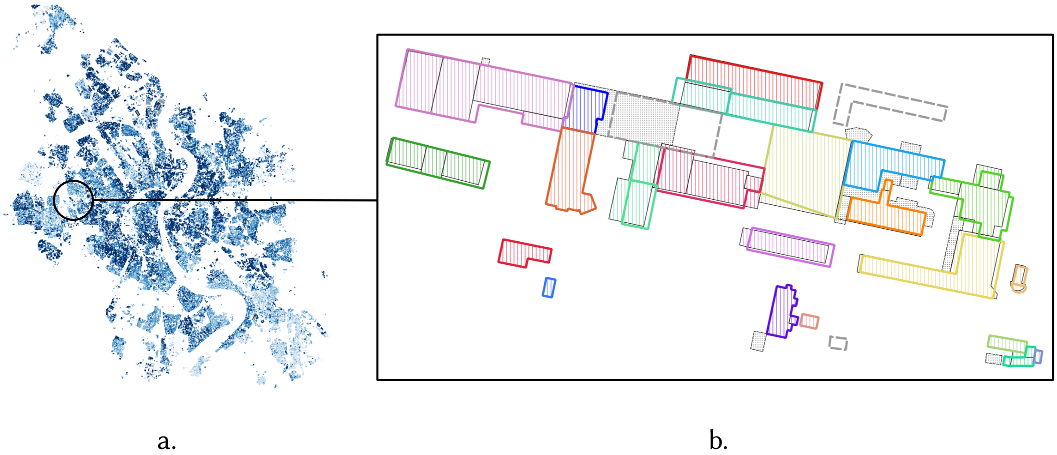

Figure 4 An optimal $m{:}n$ matching of `Cologne` for $\lambda = 0.4$. (a.) ALKIS buildings are displayed in grey, OSM buildings in blue gradients. The darker the blue color, the higher $\sigma(\mu)$ of the match μ the polygon is included in. (b.) The excerpt shows the $m{:}n$ matching with one color per match, where ALKIS buildings are displayed as a vertical line pattern and OSM buildings as their boundary. Unmatched polygons of both data sets are shown in grey.

Table 1 The sizes of the data sets used in our experiments.

City	# polygons ALKIS	# polygons OSM	# connected components of G_{int}
Bonn	119,300	97,284	40,928
Cologne	301,355	303,173	86,596

6 Experiments

To evaluate our model and algorithms with real-world data, we implemented the algorithm presented in Section 5 in C++ and provide the code in the referenced repository. For geometrical operations, we used the CGAL library [19]. The graph operations were done using the boost library and the ILP was solved with the Gurobi solver [6]. We pre-computed the connected components of the intersection graph using QGIS [16]. Before discussing the results of the runs, we want to give a brief overview of the implementation details.

Data and Experimental Settings

For our evaluation, we used German cadastral data ALKIS and the open-source data of OpenStreetMap (in the following OSM) of the cities of Bonn, Germany, and Cologne, Germany, which we call `Bonn` and `Cologne`, respectively. The OSM data sets contain all elements flagged as `building` in the geographical area of Bonn and Cologne. For the specific sizes of the datasets, see Table 1. For illustration of the datasets see Figure 1 (excerpt of `Bonn`) and Figure 4 (a.) (`Cologne`). For an exemplary result of an $m{:}n$ matching see Figure 4 (b.).

We performed experiments for $\lambda \in \{0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$. We limited the execution time per construction of the candidate graph of a connected component to 1000s. When the limit is reached, the construction stops and we solve the ILP on the candidate graph computed so far.

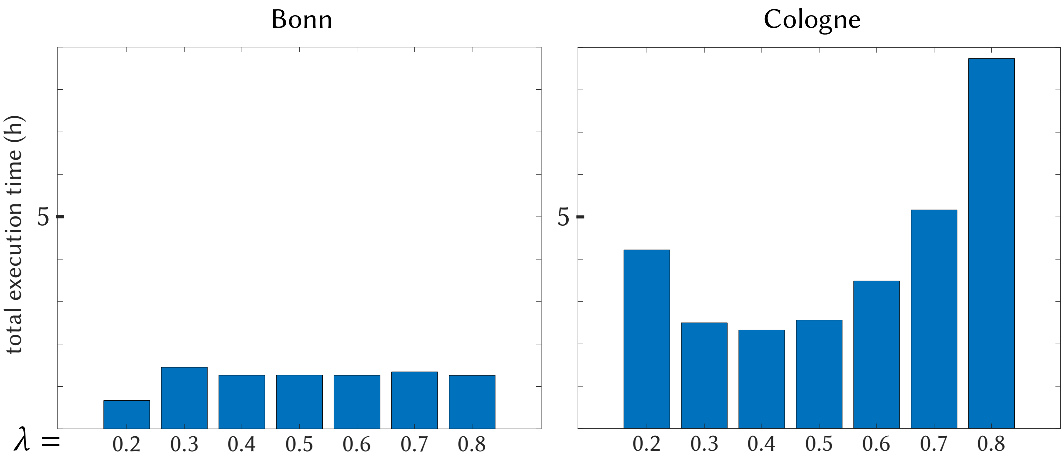

Figure 5 The total running times of Algorithm 1 after pre-computing the connected components of `Bonn` and `Cologne`.

Results: Running Time

We solved `Bonn` for every value of λ in at most $\approx$ 1.5h (max 221s per component). Among the 86,596 components of `Cologne`, we could not solve 1–12 components within the time limit, depending on λ. Figure 5 shows the running time for the different values of λ for `Bonn` and `Cologne`. Generally, a larger λ leads to more sets of polygons fulfilling Lemma 4 (Polygons in the same match) as condition (3) gets less strict. On the other hand, smaller λ favor the application of Lemma 7 (1:1 Match), because the sum of two matches exceeds the sum of one big match more easily. This leads to λ close to 0.5 striking a good balance between the applicability of both properties and hence the best running time. Figure 6 shows the impact of Lemma 4 and Lemma 7 on the average number of buildings of the connected components. The combination of Lemma 4 and Lemma 7 allows us to reduce the average number of buildings of the connected components significantly. Often, the simplification solves the m:n matching of a connected component entirely. These cases do not require the construction of the candidate graph and running the ILP and we treat their sizes as 0 in our statistic. Hence, the average size after the simplification can be less than 1. As the average number of buildings of a connected component corresponds to the average number of vertices of its intersection graph, the simplification reduces the running time tremendously.

Challenging instances. As described before, a few components of `Cologne` were not solved in the preset time limit. One exemplary component consists of 57 ALKIS- and 42 OSM-polygons (Figure 7). Here, the maps differ severely: (i) the outline of the components of ALKIS and OSM differ, and (ii) multiple OSM-building intersect multiple ALKIS-buildings. Hence, we can apply our observations and lemmas only a few times. The algorithm stopped during the construction of the candidate graph and it consisted already of 32,964 nodes and 62,323 edges. In application, we propose to limit the maximum number of represented polygons per cumulative vertex to a fixed value k (in this example, $k = 10$ would suffice and lead to a running time of $\approx$ 20s with a graph of 8,449 nodes and 36,482 edges with $\lambda = 0.4$).

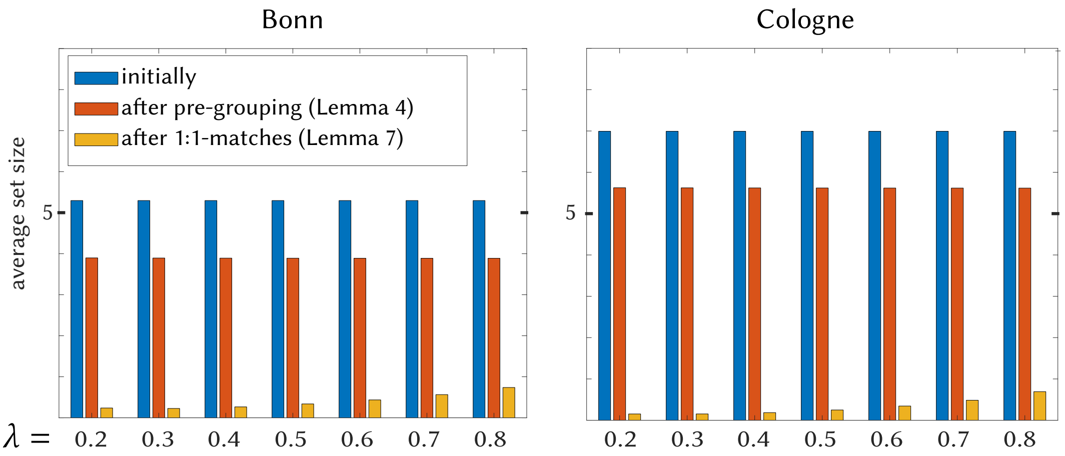

Figure 6 The average (= arithmetic mean) size (= number of vertices) of a component occurring in the data sets before and after the application of our observations.

Results: Reasonability

We analyzed the size of m and n of the matches included in optimal $m{:}n$ matchings for $\lambda \in \{0.4, 0.6, 0.8\}$ (Figure 8). Larger values of λ favor solutions with less matches and higher qualities. This is due to our choice of $\sigma(M)$. The more matches M includes, the more often λ gets subtracted in $\sigma(M)$. Hence, the larger λ gets, the more the total number of matches is penalized. On the contrary, small λ lead to solutions including many matches with few buildings and often less quality (see Figure 1 (b.) as an example for a possible match using a small λ).

Results: Comparison to 1:1 Matching

As shown in Figure 8, the optimal $m{:}n$ matching contains matches with $m > 1$ and $n > 1$. This validates our approach of using an $m{:}n$ matching instead of a 1:1 matching. To rate the improvement, we compared our approach to the optimal 1:1 matching $M_{1:1}$ maximizing $\sigma(M_{1:1})$ as a baseline. We obtained qualities in the range of $84\% - 87\%$ for the different λ values with respect to optimal $m{:}n$ matching qualities. For smaller λ, the similarity is higher. Further, we compared the number of matches for optimal 1:1- and $m{:}n$ matchings. It showed that while for $\lambda = 0.3$ we have $\approx 1\%$ less matches in the 1:1 matching than in the $m{:}n$ matching, this difference grows to $\approx 9\%$ for $\lambda = 0.7$. This aligns well with the observation stated in the last paragraph that higher λ favor larger matches, which cannot be formed with 1:1 matching.

7 Conclusion and Outlook

We presented the first optimization approach for finding an $m{:}n$ matching of two sets of polygons. We introduced a natural quality measure of a $m{:}n$ matching based on the established Jaccard index. We formalized the problem of finding an optimal $m{:}n$ matching w.r.t. this measure. A naive approach via integer programming for solving this problem is not feasible for real-world data and average computing systems. We presented structural properties of any optimal solution that allows us to make the computation feasible. We implemented the algorithm and evaluated it on building footprint data of the cities of Bonn and Cologne.

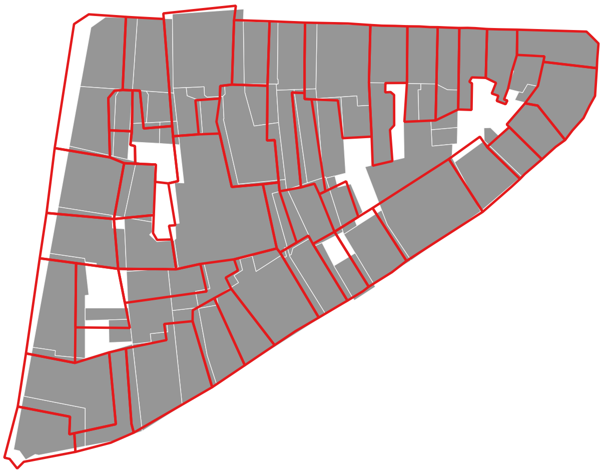

Figure 7 The instance that could not be solved optimally within our chosen execution time threshold of 1000s. The ALKIS data is shown in grey, and OSM in red.

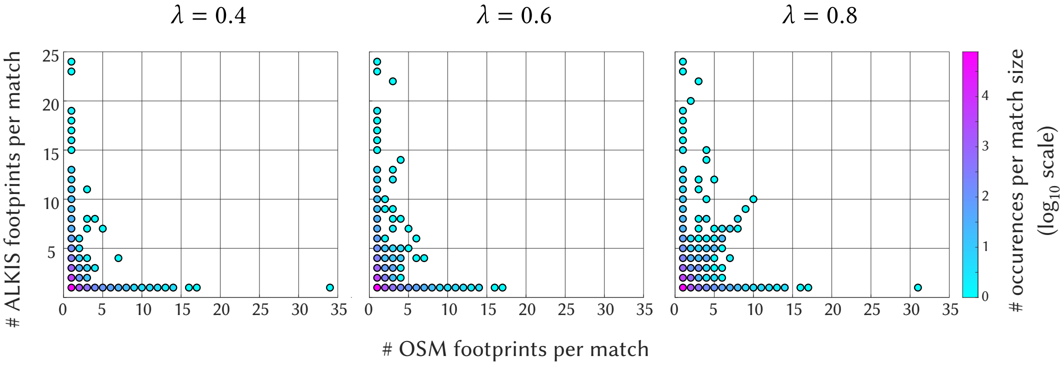

Figure 8 The numbers of occurrences per match composition w.r.t. the number of included OSM and ALKIS polygons of `Bonn`. The numbers are represented by a color gradient using a logarithmic scale.

Our m:n matching approach allows combining information from two data sets as well as the evaluation of the quality of one data set with respect to the other one. Alongside the m:n matching itself, our algorithm implicitly assigns each match a quality index. This enables detailed analysis as well as visualization of the correspondences. These tasks are important in real-world applications, as the total amount of data continuously increases, making combination as well as evaluation necessary. Our algorithm applies for any application working with polygons. Hence, it is not limited to building footprints.

Although we were able to find multiple properties of optimal solutions that reduced the running time, a few instances could not be solved within the chosen time limit. Hence, investigating structural properties for an exact solution or considering an approximation are open research topics. It could be possible to simplify the graph even more within approximation guarantees. This could decrease the running time with possibly only a negligible loss of solution quality. It would also be interesting to explore other objectives for a matching that do not scale linearly w.r.t. the Jaccard index. Furthermore, one could explore how the $m{:}n$ matching can be used for further processing. For example, for merging the geometries of two maps or combining the polygon attributes, the $m{:}n$ matching can serve as a starting point for additional computations.

References

1 Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, 2016. doi:10.1109/ICIP.2016.7533003.

2 Bezirksregierung Köln. ALKIS (Amtliches Liegenschaftskatasterinformationssystem). Data accessed from ALKIS via https://www.opengeodata.nrw.de/produkte/geobasis/lk/akt/gru_xml/, 2023. Accessed: 01.03.2024.

3 Stefan Canzar, Khaled Elbassioni, Gunnar Klau, and Julián Mestre. On tree-constrained matchings and generalizations. *Algorithmica*, 71:98–109, July 2011. doi:10.1007/978-3-642-22006-7_9.

4 Muhammed Fatih Demirci. *Many-to-many feature matching for structural pattern recognition.* PhD thesis, Drexel University, USA, 2005. AAI3194363.

5 Hongchao Fan, Alexander Zipf, Qing Fu, and Pascal Neis. Quality assessment for building footprints data on openstreetmap. *International Journal of Geographical Information Science*, 28(4):700–719, 2014. doi:10.1080/13658816.2013.867495.

6 Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023. URL: https://www.gurobi.com.

7 Robert Hecht, Carola Kunze, and Stefan Hahmann. Measuring completeness of building footprints in openstreetmap over space and time. *ISPRS International Journal of Geo-Information*, 2(4):1066–1091, 2013. doi:10.3390/ijgi2041066.

8 Shahab Jozdani and Dongmei Chen. On the versatility of popular and recently proposed supervised evaluation metrics for segmentation quality of remotely sensed images: An experimental case study of building extraction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 160:275–290, 2020. doi:10.1016/j.isprsjprs.2020.01.002.

9 Richard Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations*, 40:85–103, January 1972. doi:10.1007/978-3-540-68279-0_8.

10 Jung Ok Kim, Kiyun Yu, Joon Heo, and Won Hee Lee. A new method for matching objects in two different geospatial datasets based on the geographic context. *Computers & Geosciences*, 36(9):1115–1122, 2010. doi:10.1016/j.cageo.2010.04.003.

11 Lingjia Liu, Xinyan Zhu, Daoye Zhu, and Ding Xiaohui. M:n object matching on multiscale datasets based on mbr combinatorial optimization algorithm and spatial district. *Transactions in GIS*, 22, November 2018. doi:10.1111/tgis.12488.

12 Fabian Müller, Ionuț Iosifescu, and Lorenz Hurni. Assessment and visualization of osm building footprint quality. In *Proceedings of the 27th International Cartographic Conference (ICC 2015)*, Rio de Janeiro, Brazil, August 2015.

13 Samuel Murray. *Real-Time Multiple Object Tracking - A Study on the Importance of Speed.* Degree project, KTH Royal institute of technology school of computer science and communication, Stockholm, Sweden, September 2017.

14 Meghana Nasre, Prajakta Nimbhorkar, Keshav Ranjan, and Ankita Sarkar. Popular critical matchings in the many-to-many setting. *Theoretical Computer Science*, 982:114281, 2024. doi:10.1016/j.tcs.2023.114281.

15 OpenStreetMap Contributors. OpenStreetMap. https://www.openstreetmap.org, 2004–*present*. Accessed: 01.03.2024.

16 QGIS Development Team. *QGIS Geographic Information System.* QGIS Association, URL: https://www.qgis.org.

17 Juan J. Ruiz-Lendínez, Manuel A. Ureña-Cámara, and Francisco J. Ariza-López. A polygon and point-based approach to matching geospatial features. *ISPRS International Journal of Geo-Information*, 6(12), 2017. doi:10.3390/ijgi6120399.

18 Ali Shokoufandeh, Yakov Keselman, Fatih Demirci, Diego Macrini, and Sven Dickinson. *Many-to-Many Feature Matching in Object Recognition*, pages 107–125. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. doi:10.1007/11414353_8.

19 The CGAL Project. *CGAL User and Reference Manual*. CGAL Editorial Board, 5.6 edition, 2023. URL: https://doc.cgal.org/5.6/Manual/packages.html.

20 Matthias P. Wagner and Natascha Oppelt. Extracting agricultural fields from remote sensing imagery using graph-based growing contours. *Remote Sensing*, 12(7), 2020. doi:10.3390/rs12071205.

21 Volker Walter. *Matching of spatial data - on the example of the datamodels ATKIS and GDF*. Phd thesis, German Geodetic Commission, Munich, Germany, 1997.

22 Volker Walter and Dieter Fritsch. Matching spatial data sets: a statistical approach. *International Journal of Geographical Information Science*, 13(5):445–473, 1999. doi:10.1080/136588199241157.

23 Xiaofang Wang, Yu Zang, Yiping Chen, Cheng Wang, and Jonathan Li. Rural road networks matching via extending line. In *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 7419–7422, 2018. doi:10.1109/IGARSS.2018.8519400.

24 Kiyun Yu Yong Huh, Jiyoung Kim and Sunghwan Cho. M:n object matching between image and map object data sets by means of latent semantic analysis. *International Journal of Remote Sensing*, 35(18):6799–6814, 2014. doi:10.1080/01431161.2014.965286.

25 Mikhail Zaslavskiy, Francis Bach, and Jean-Philippe Vert. Many-to-many graph matching: A continuous relaxation approach. In *Machine Learning and Knowledge Discovery in Databases*, pages 515–530, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.