



Dynamische Beschriftung von Straßen

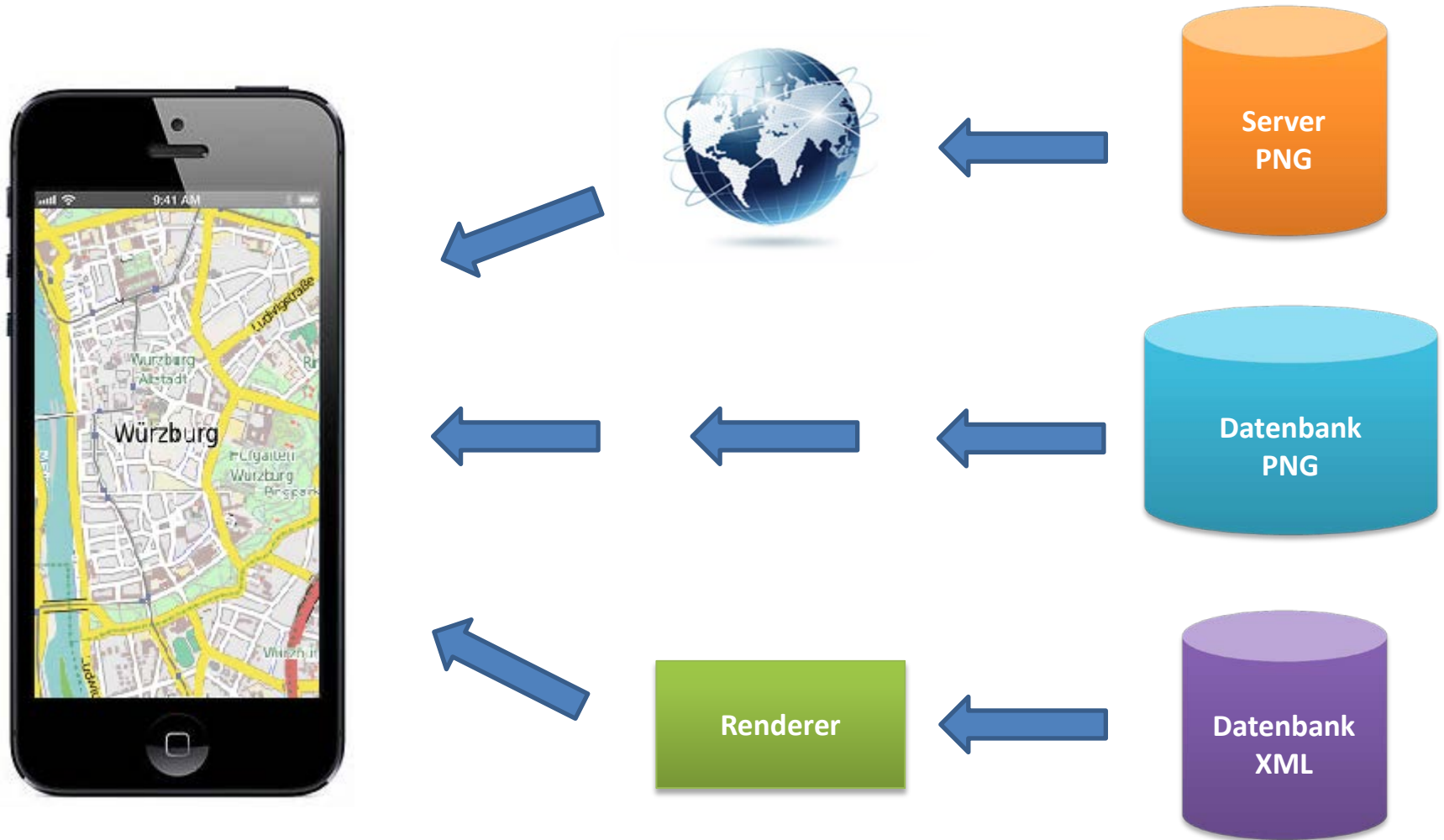
Sergiy Pakholchak
Bachelorkolloquium 2013

Inhalt

- ▶ Einführung
 - OpenStreetMap
 - Mapsforge
- ▶ Kandidatengenerierung
- ▶ Beschriftungsalgorithmus
 - Schnittpunktermittlung
 - Cache
 - 3-Regel-Algorithmus
- ▶ Experimente

Einführung

Darstellung von Karten auf mobilen Endgeräten



Beschriftung von Straßen

▶ Input

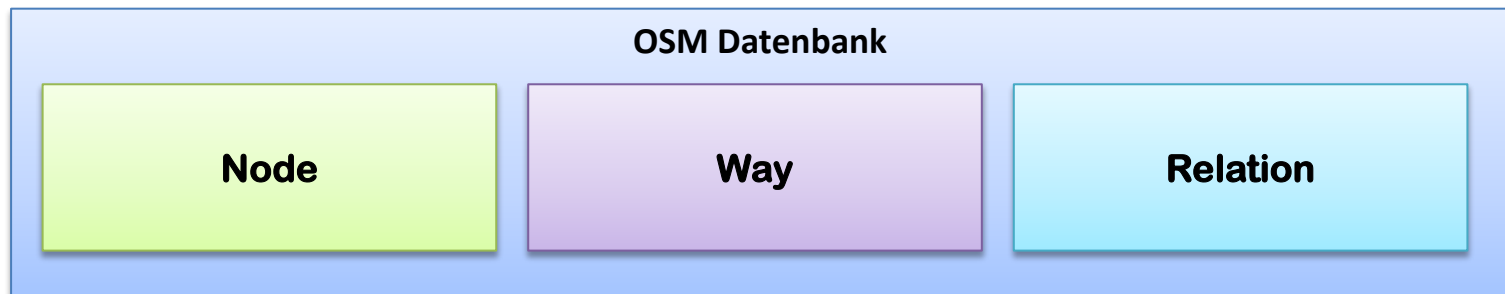
- OSM-Rohdaten in XML-Form
- Polygonzug von Geokoordinaten, die eine Straße bilden
- Straßennamen

▶ Zielsetzung

- Überschneidungen vermeiden (harte Beschränkung)
- so viele Straßen wie möglich beschriften (weiche Beschränkung)

OpenStreetMap

- ▶ Ein freies Projekt, das von Steve Coast gegründet (2004)
- ▶ Datenbank von Geodaten und Metainformation
- ▶ XML-Format
 - einfach strukturiert
 - Element *Tag* zur Angaben von Metainformation



OpenStreetMap

```
<node id="116117689" version="2" timestamp="2011-02-06T22:01:50Z" uid="57645"
user="KartoGrapHiti" changeset="7210593" lat="49.7972401" lon="9.9372145" />
<node id="50487231" version="1" timestamp="2007-09-13T10:29:25Z" uid="7197"
user="user_7197" changeset="422370" lat="49.7973092" lon="9.9373836" />
```

```
<way id="30321025" version="2" timestamp="2010-04-10T17:35:19Z"
uid="57158" user="kay_D" changeset="4384942">
  <nd ref="116117689" />
  <nd ref="50487231" />
  <nd ref="691752832" />
  <nd ref="297734634" />
  <nd ref="50487234" />
  <tag k="abutters" v="retail" />
  <tag k="highway" v="residential" />
  <tag k="maxspeed" v="30" />
  <tag k="name" v="Semmelstrasse" />
</way>
```

OpenStreetMap

```
<node id="116117689" version="2" timestamp="2011-02-06T22:01:50Z" uid="57645"  
user="KartoGrapHiti" changeset="7210593" lat="49.7972401" lon="9.9372145" />  
<node id="50487231" version="1" timestamp="2007-09-13T10:29:25Z" uid="7197"  
user="user_7197" changeset="422370" lat="49.7973092" lon="9.9373836" />
```

```
<way id="30321025" version="2" timestamp="2010-04-10T17:35:19Z"  
uid="57158" user="kay_D" changeset="4384942">  
  <nd ref="116117689" />  
  <nd ref="50487231" />  
  <nd ref="691752832" />  
  <nd ref="297734634" />  
  <nd ref="50487234" />  
  <tag k="abutters" v="retail" />  
  <tag k="highway" v="residential" />  
  <tag k="maxspeed" v="30" />  
  <tag k="name" v="Semmelstrasse" />  
</way>
```

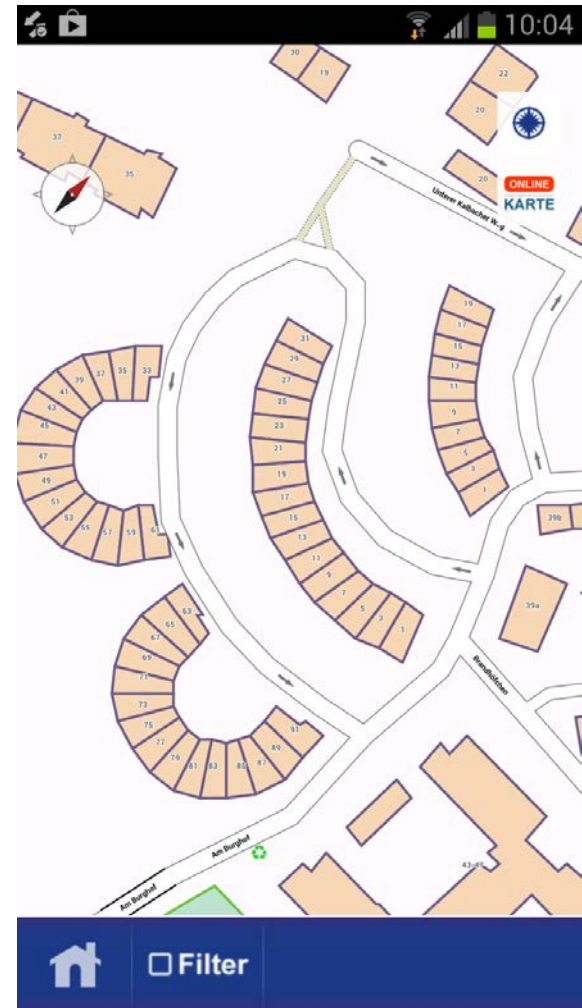
Segment



- ▶ Jede Straße besteht aus einem oder mehreren *Segmenten*.
- ▶ Länge des Segmentes ändert sich in Abhängigkeit von der Zoomstufe

Mapforge

- ▶ Frei zugängliche OSM-Rendererer für die Java-Plattform Android
- ▶ Beschriftung der Straßen nur an geraden Segmenten
- ▶ Vorteil:
 - Überschneidungen werden ohne zusätzlichen Rechenaufwand vermieden
- ▶ Nachteile:
 - Straßenbeschriftung erst in höheren Zoomstufen möglich
 - kurvige Straßen können nicht beschriftet werden



Kandidatengenerierung

Kandidat

- ▶ Möglich Stelle für die Beschriftung einer Straße
- ▶ Eine Straße hat mehrere Kandidaten



Kandidatengenerierung

Kandidat

- ▶ Möglich Stelle für die Beschriftung einer Straße
- ▶ Eine Straße hat mehrere Kandidaten



Kandidatengenerierung

Kandidat

- ▶ Möglich Stelle für die Beschriftung einer Straße
- ▶ Eine Straße hat mehrere Kandidaten



Kandidatengenerierung

Kandidat

- ▶ Möglich Stelle für die Beschriftung einer Straße
- ▶ Eine Straße hat mehrere Kandidaten



Kandidatengenerierung

Kandidat

- ▶ Möglich Stelle für die Beschriftung einer Straße
- ▶ Eine Straße hat mehrere Kandidaten
- ▶ Prioritäten vergeben
 - gute optische Darstellung
 - möglichst mittig
 - scharfen Kurven vermeiden
 - vertikale Beschriftungen vermeiden
 - weniger Überschneidungen mit anderen Straßen



Kandidatengenerierung

Objekt



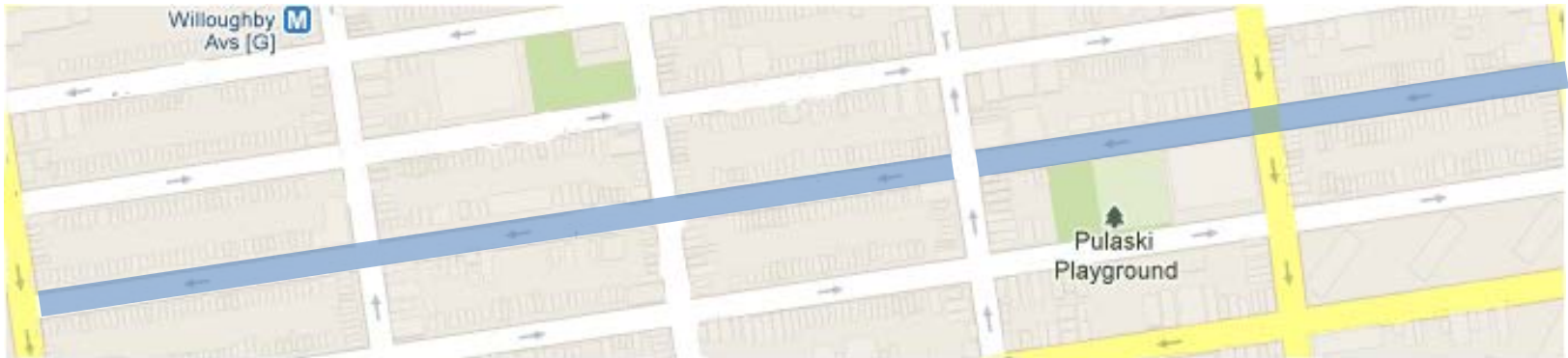
Kandidatengenerierung

Objekt



Kandidatengenerierung

Objekt



Kandidatengenerierung

Objekt



Kandidatengenerierung

Objekt



Kandidatengenerierung

Objekt



Kandidatengenerierung

Objekt

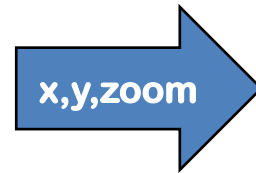


- ▶ Hat einen oder mehrere Kandidaten
 - ein Kandidat — falls ein Segment existiert, das groß genug für eine Beschriftung ist
 - ein Kandidat — falls keine Überschneidungen mit Nachbarstraßen existieren
- ▶ Nach der Anwendung des Algorithmus bleibt höchstens ein Kandidat pro Objekt übrig

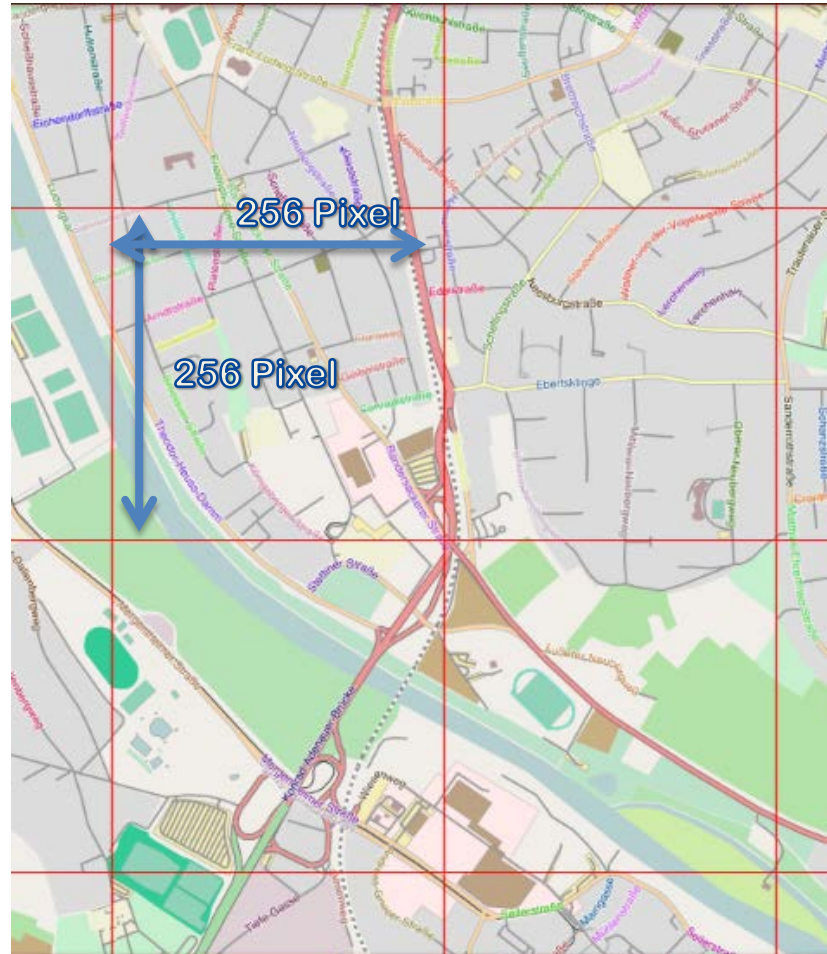
Schnittpunktermittlung

- ▶ Aufwändig für mobile Endgerät
- ▶ Realisiert durch einfachen Algorithmus mit Laufzeit $O(n^2)$
- ▶ Schnittpunkte können alternativ mit dem Algorithmus von Bentley und Ottmann ermittelt werden
 - verbessert die Berechnungszeit zu $O((k + n)\log n)$
wobei n die Anzahl der Strecken und k die Anzahl der Schnittpunkte ist
- ▶ Schnittpunkte sind nötig:
 - um Kandidaten zu generieren
 - um Prioritäten zu vergeben
 - für den eigentlichen Algorithmus

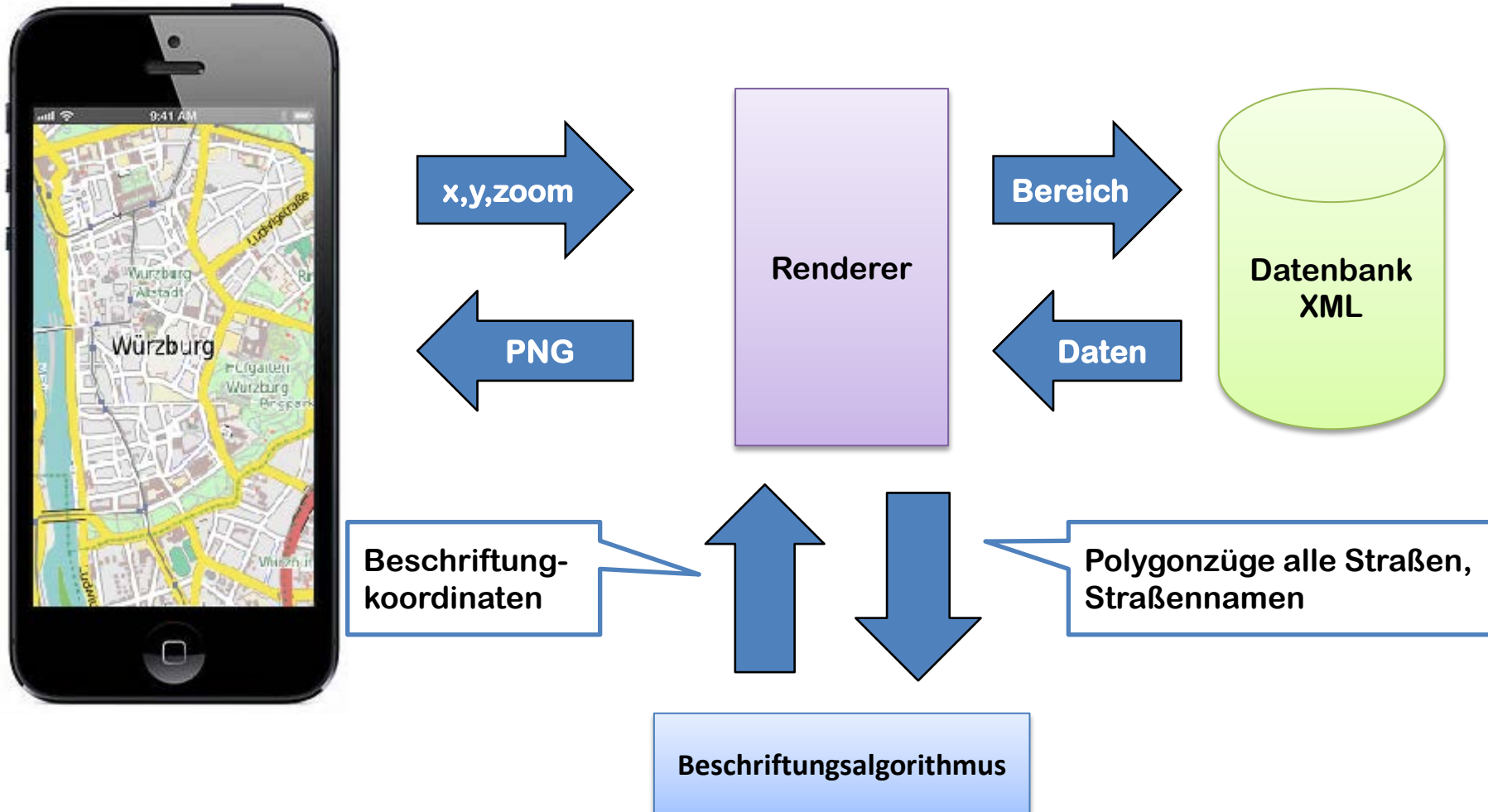
Beschriftungsalgorithmus



Beschriftungsalgorithmus



Beschriftungsalgorithmus

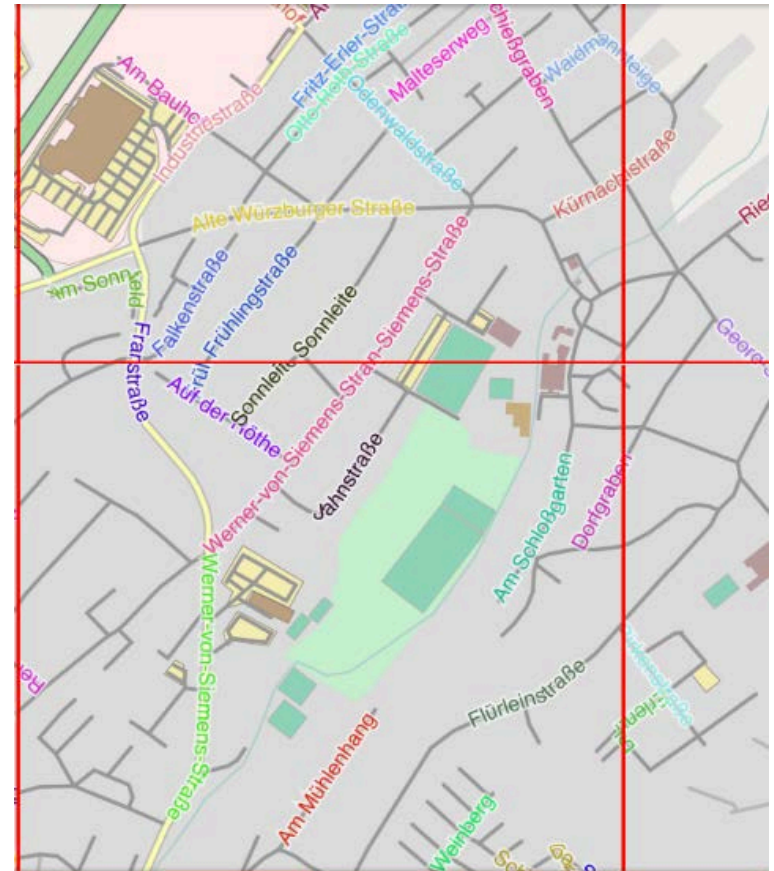


Beschriftungsalgorithmus

- ▶ Input:
 - Polygonzüge aller Straßen
 - Straßennamen (Schriftgröße, Schriftart)
- ▶ Ermittlung von Objekten
 - jede Straße wird in Abhängigkeit von ihrer Länge in Objekte aufgeteilt
- ▶ Schnittpunktermittlung
- ▶ Kandidatengenerierung
 - vorhandene Objekte aus dem Cache holen
 - neue Kandidaten generieren
- ▶ Kandidatenauswahl
 - 3-Regel-Algorithmus anwenden
- ▶ Speichern der berechneten Beschriftungen im Cache

Cache

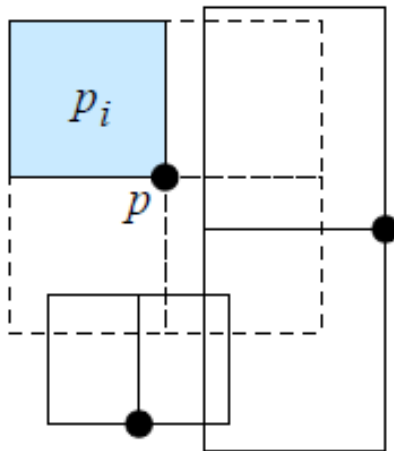
- ▶ Objekte/Kandidaten verlaufen kachelübergreifend
- ▶ Objekte werden im Cache gespeichert
- ▶ Gleiche Objekte von verschiedenen Kacheln bekommen gleiche Beschriftung
- ▶ spart Rechenzeit



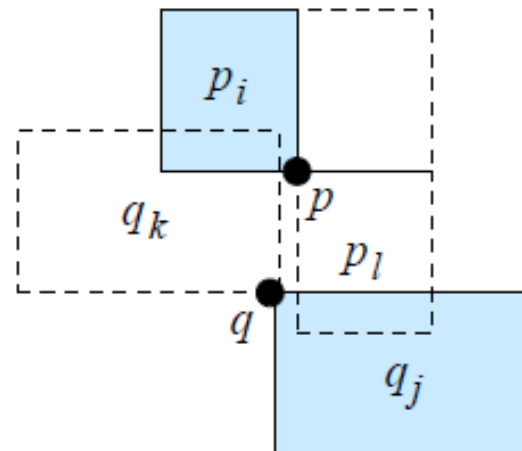
3-Regel-Algorithmus

Der Algorithmus wurde von Wagner et al. entwickelt und besteht aus zwei Phasen

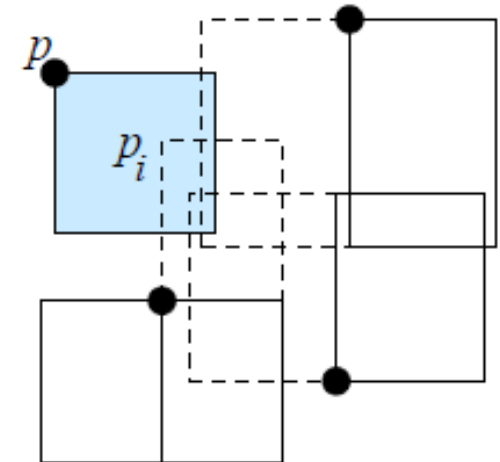
Phase 1: Für alle Kandidaten werden die drei folgende Regeln angewendet



Regel 1



Regel 2



Regel 3

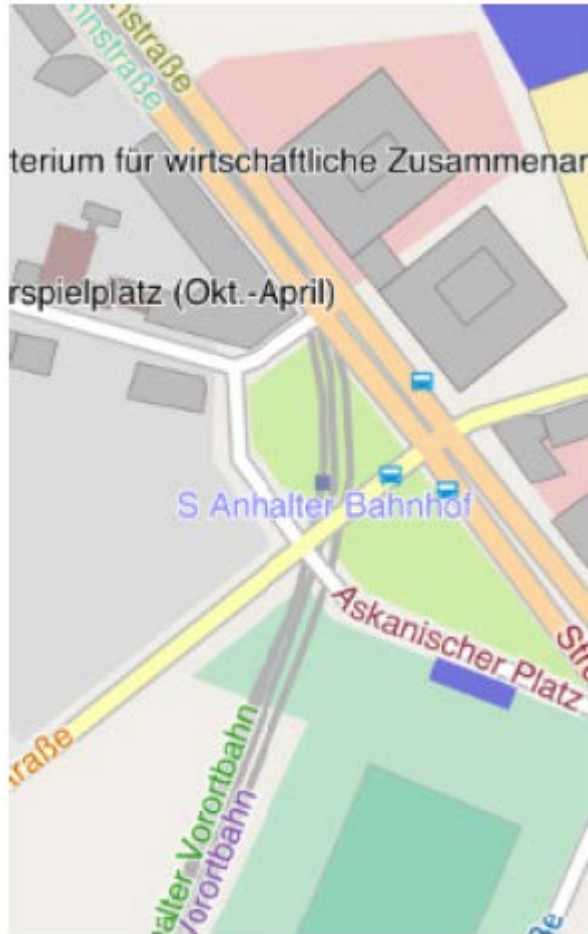
3-Regel-Algorithmus

Falls es Objekte gibt, die mehr als einen Kandidaten besitzen

Phase 2:

- ▶ Man durchläuft alle Objekte mit größter Anzahl an Kandidaten
- ▶ Von diesen Objekten werden die ungünstigsten Kandidaten entfernt
- ▶ Wie in Phase 1 werden die drei Regeln rekursiv für die Nachbarn diese Objekte durchgeführt
- ▶ Laufzeit $O(k + n \log n)$

POI-Beschriftungen



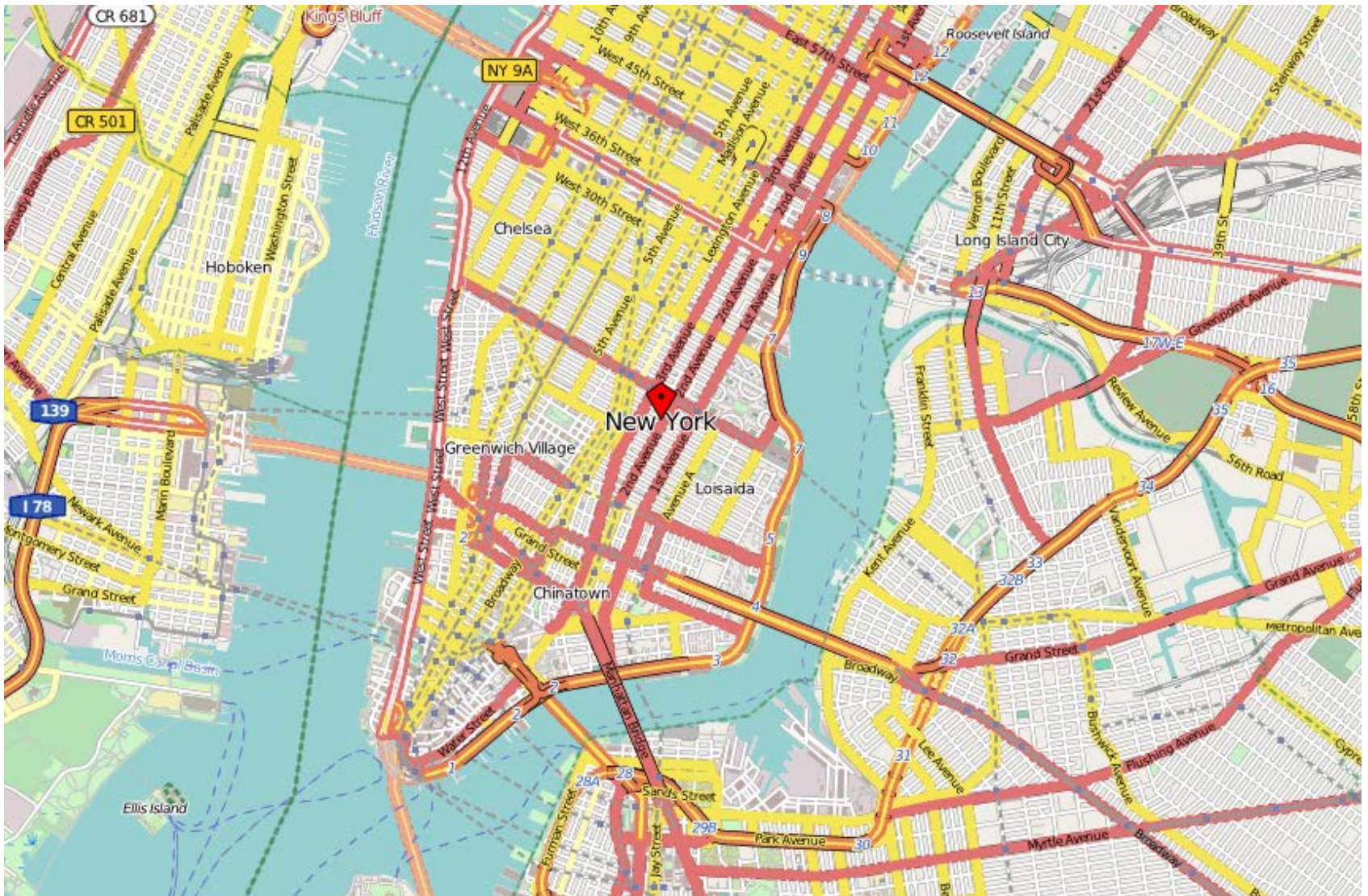
Vermeidung von Überschneidungen bei Straßennamen und POI-Beschriftungen

- ▶ Eine POI-Beschriftung wird als ein Objekt, das aus einem Segment und einem Kandidat besteht, betrachtet
- ▶ Diese Objekte werden bei dem 3-Regel-Algorithmus mit berücksichtigt
- ▶ Kandidat bekommt höchste Priorität und wird immer als Lösung des Algorithmus betrachtet

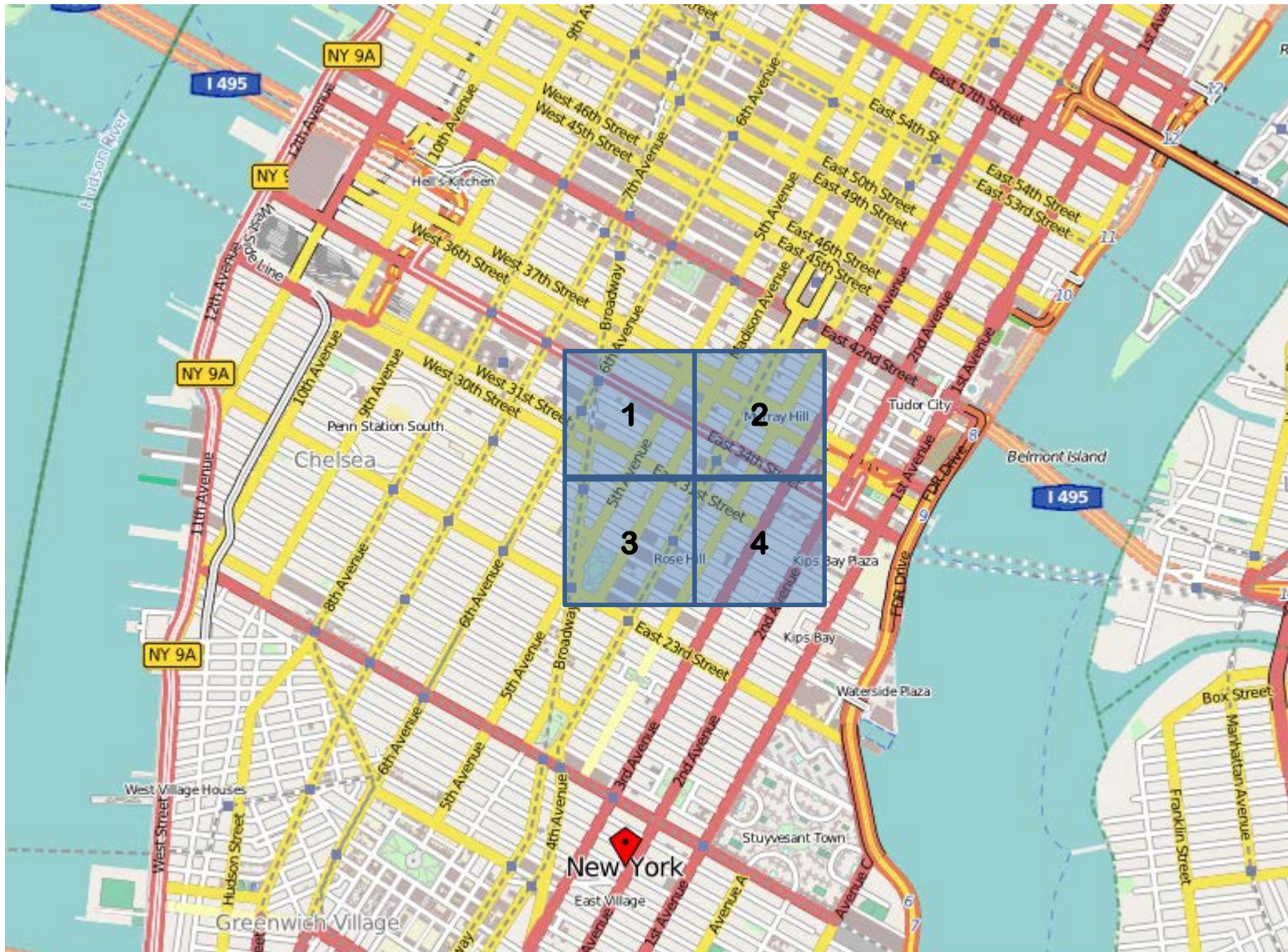
Experimente

- ▶ Experimente werden mit dem Renderer libosmscout durchgeführt
- ▶ Testgerät iPhone4 mit Betriebssystem iOS Version 6.0.
- ▶ Laufzeit $O(k + n \log n)$

Experimente

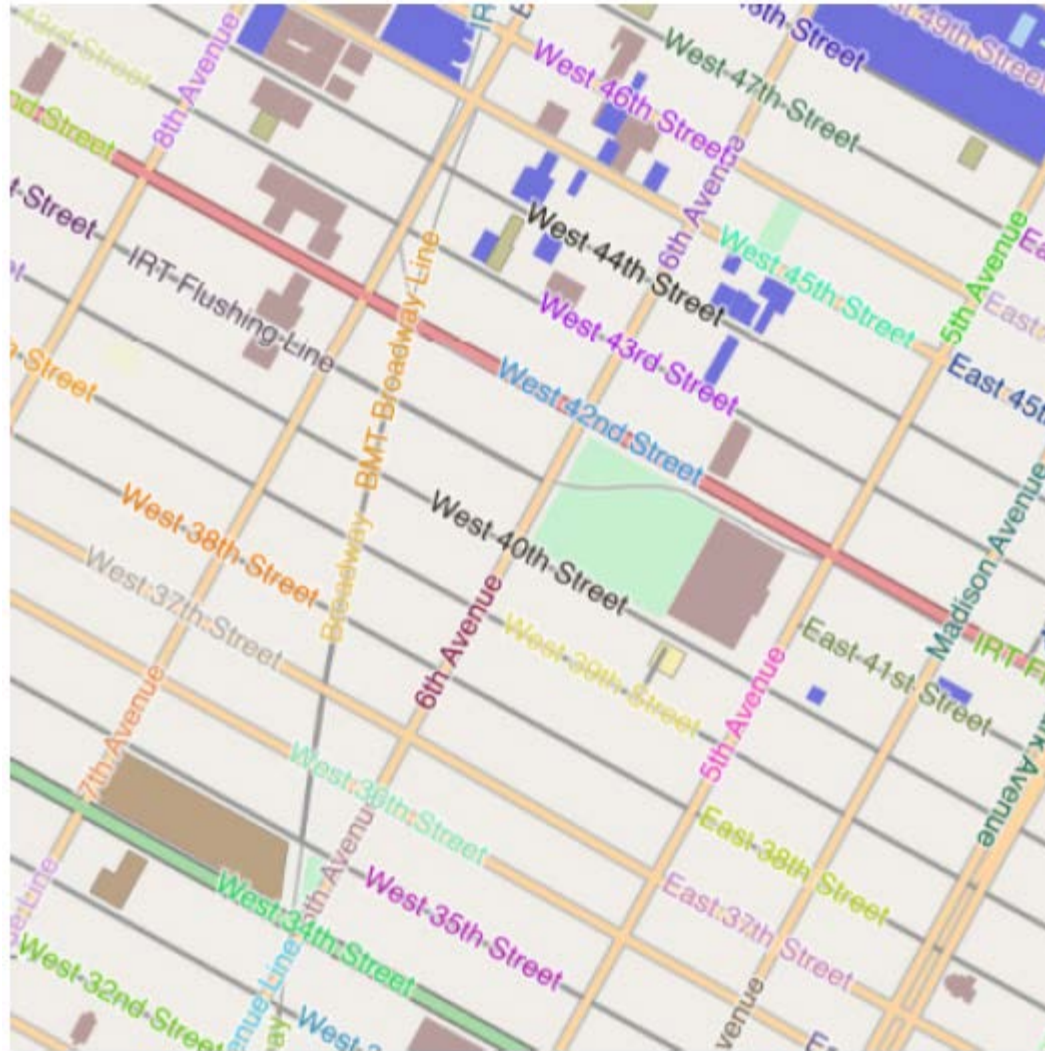


Experimente



Experimente

Kachel 1



Experimente

► Ergebnisse des 3-Regel-Algorithmus ohne Cache

Kachel	Dauer in Sek für Anzahl von Kandidaten						Objekten	besch. Str.
	reduzierte Kandidaten			alle Kandidaten				
	Anzahl	b. O.	Dauer	Anzahl	b. O.	Dauer		
1	413	87	1,14	752	85	1,32	89	41 von 43
2	558	104	1,17	942	102	1,47	109	55 von 57
3	641	113	1,45	929	112	1,89	119	59 von 60
4	557	91	1,15	749	90	1,23	95	55 von 57

► Ergebnisse des 3-Regel-Algorithmus mit Cache

Kachel	Dauer in Sek für Anzahl von Kandidaten						Objekten	besch. Str.
	reduzierte Kandidaten			alle Kandidaten				
	Anzahl	b. O.	Dauer	Anzahl	b. O.	Dauer		
1	413	87	1,14	752	85	1,32	89	41 von 43
2	274	104	0,82	532	102	0,99	109	55 von 57
3	309	113	1,15	420	112	1,41	119	59 von 60
4	197	91	0,71	277	90	0,85	95	55 von 57

Experimente

► Ergebnisse des 3-Regel-Algorithmus ohne Cache

Kachel	Dauer in Sek für Anzahl von Kandidaten						Objekten	besch. Str.
	reduzierte Kandidaten			alle Kandidaten				
	Anzahl	b. O.	Dauer	Anzahl	b. O.	Dauer		
1	413	87	1,14	752	85	1,32	89	41 von 43
2	558	104	1,17	942	102	1,47	109	55 von 57
3	641	113	1,45	929	112	1,89	119	59 von 60
4	557	91	1,15	749	90	1,23	95	55 von 57

► Ergebnisse des 3-Regel-Algorithmus mit Cache

Kachel	Dauer in Sek für Anzahl von Kandidaten						Objekten	besch. Str.
	reduzierte Kandidaten			alle Kandidaten				
	Anzahl	b. O.	Dauer	Anzahl	b. O.	Dauer		
1	413	87	1,14	752	85	1,32	89	41 von 43
2	274	104	0,82	532	102	0,99	109	55 von 57
3	309	113	1,15	420	112	1,41	119	59 von 60
4	197	91	0,71	277	90	0,85	95	55 von 57

Experimente

► Ergebnisse des 3-Regel-Algorithmus ohne Cache

Kachel	Dauer in Sek für Anzahl von Kandidaten						Objekten	besch. Str.
	reduzierte Kandidaten			alle Kandidaten				
	Anzahl	b. O.	Dauer	Anzahl	b. O.	Dauer		
1	413	87	1,14	752	85	1,32	89	41 von 43
2	558	104	1,17	942	102	1,47	109	55 von 57
3	641	113	1,45	929	112	1,89	119	59 von 60
4	557	91	1,15	749	90	1,23	95	55 von 57

► Ergebnisse des 3-Regel-Algorithmus mit Cache

Kachel	Dauer in Sek für Anzahl von Kandidaten						Objekten	besch. Str.
	reduzierte Kandidaten			alle Kandidaten				
	Anzahl	b. O.	Dauer	Anzahl	b. O.	Dauer		
1	413	87	1,14	752	85	1,32	89	41 von 43
2	274	104	0,82	532	102	0,99	109	55 von 57
3	309	113	1,15	420	112	1,41	119	59 von 60
4	197	91	0,71	277	90	0,85	95	55 von 57

Experimente

► Ergebnisse des 3-Regel-Algorithmus ohne Cache

Kachel	Dauer in Sek für Anzahl von Kandidaten						Objekten	besch. Str.
	reduzierte Kandidaten			alle Kandidaten				
	Anzahl	b. O.	Dauer	Anzahl	b. O.	Dauer		
1	413	87	1,14	752	85	1,32	89	41 von 43
2	558	104	1,17	942	102	1,47	109	55 von 57
3	641	113	1,45	929	112	1,89	119	59 von 60
4	557	91	1,15	749	90	1,23	95	55 von 57

► Ergebnisse des 3-Regel-Algorithmus mit Cache

Kachel	Dauer in Sek für Anzahl von Kandidaten						Objekten	besch. Str.
	reduzierte Kandidaten			alle Kandidaten				
	Anzahl	b. O.	Dauer	Anzahl	b. O.	Dauer		
1	413	87	1,14	752	85	1,32	89	41 von 43
2	274	104	0,82	532	102	0,99	109	55 von 57
3	309	113	1,15	420	112	1,41	119	59 von 60
4	197	91	0,71	277	90	0,85	95	55 von 57

Experimente



Experimente



Experimente

- ▶ Ergebnisse des 3-Regel-Algorithmus mit und ohne Berücksichtigung von POI-Beschriftungen für ein Gebiet von Würzburg

POI	Dauer in Sek für Anzahl von Kandidaten							
	reduzierte Kandidaten				alle Kandidaten			
	Anzahl	b. O.	b. Str.	Dauer	Anzahl	b. O.	b. Str.	Dauer
ohne	1234	275	270	12,12	2420	266	262	19,16
mit	1236	271	266	12,82	2425	263	259	19,92

Fazit

- ▶ Scheinbar einfaches Thema entpuppt sich als komplexe Problemstruktur
- ▶ Gute Ergebnisse mit 3-Regel Algorithmus
- ▶ Laufzeit verbessern
- ▶ Optische Darstellung verbessern

