Ansätze zur Lokalisierung einer Openstreetmap basierten Weltkarte

Sven Geggus

Ausgangssituation

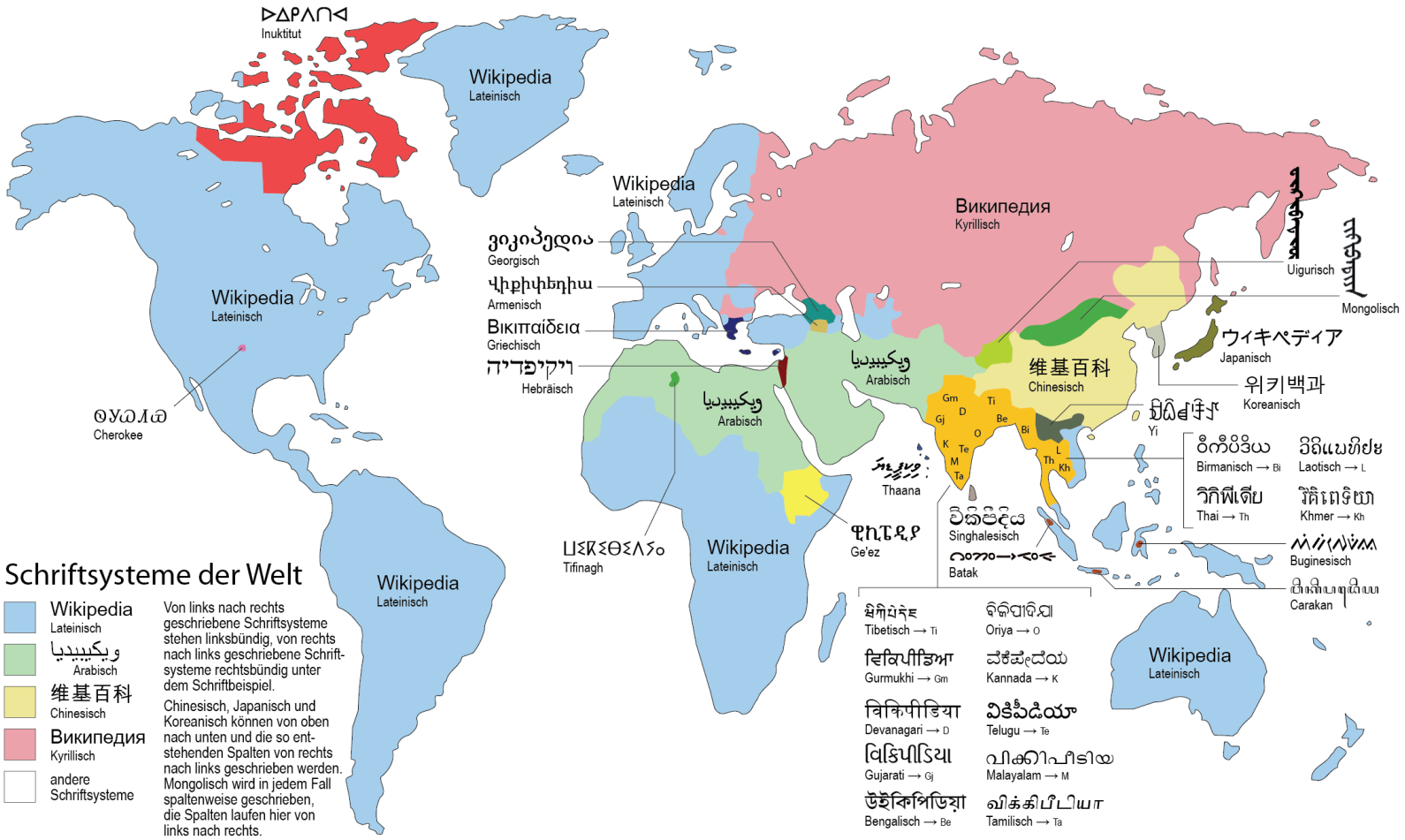
- In Gegenden, in denen nicht das lateinische Schriftsystem dominiert, sind auf Openstreetmap basierende Karten für westliche Betrachter meist unlesbar.
- Grund für die Unlesbarkeit ist die Regel, dass Namen von Objekten bei Openstreetmap in lokaler Sprache erfasst werden.
- Im Gegensatz zu herkömmlichen Geodaten enthalten viele Objekte im Openstreetmap-Datensatz jedoch zusätzlich lokalisierte Namen, die bisher in Karten noch eher selten verwendet werden.

Lokalisierte Objekte in Openstreetmap-Daten

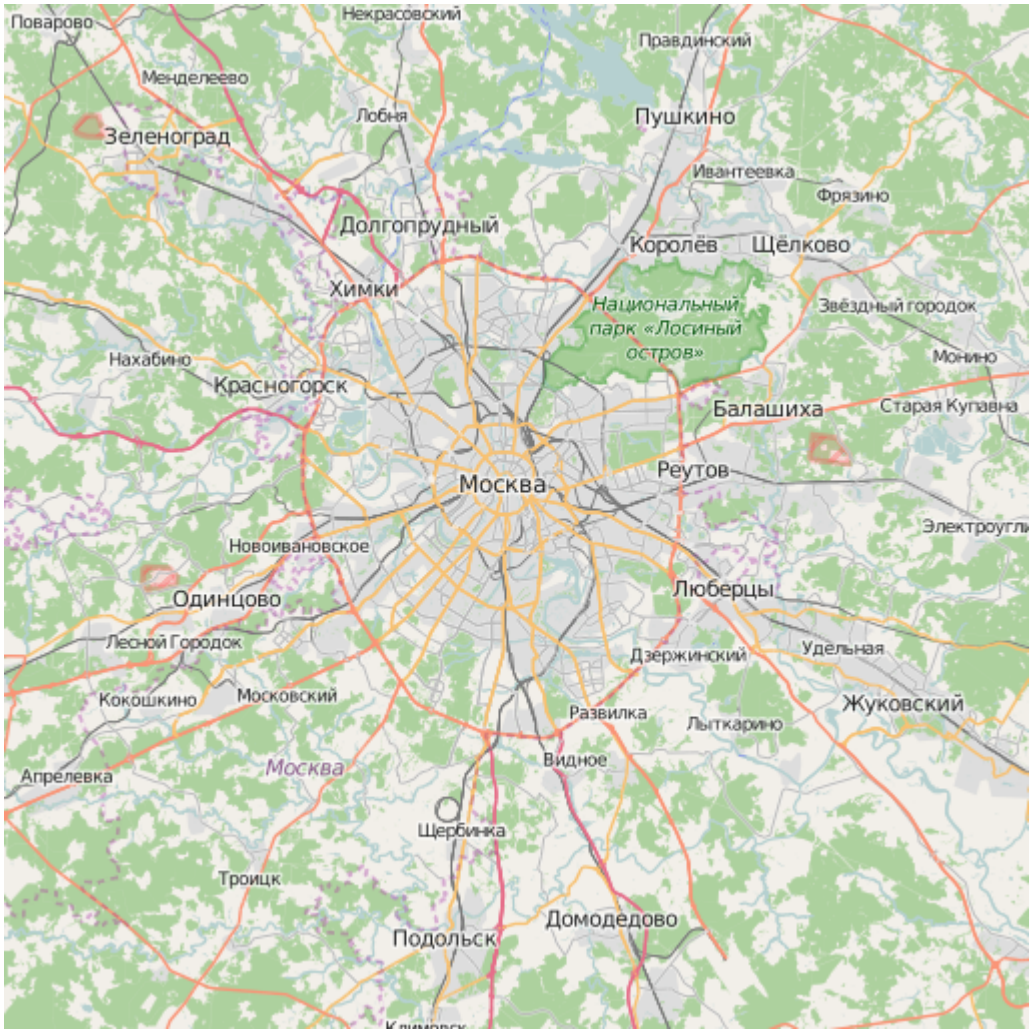
name	=> Deutschland
...	
int_name	=> Deutschland
name:de	=> Deutschland
name:en	=> Germany
name:ar	=> ألمانيا
name:ja	=> ドイツ
name:ru	=> Германия
name:is	=> גרמניה

name	=> ישראל
...	
int_name	=> Israel
name:de	=> Israel
name:en	=> Israel
name:ar	=> إسرائيل
name:ja	=> イスラエル
name:ru	=> Израиль
name:is	=> ישראל

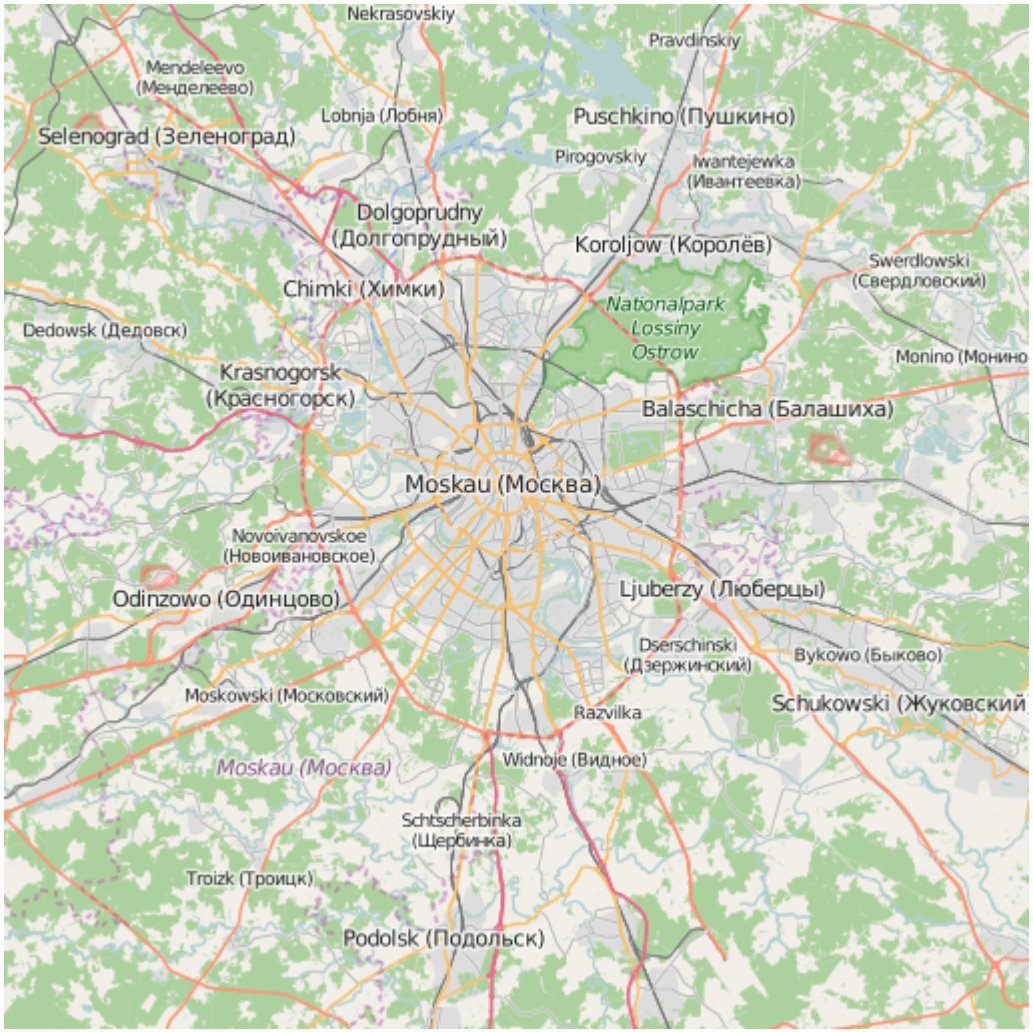
Schriftsysteme der Welt



Openstreetmap Carto Style (Original)



Openstreetmap Carto Style (lokalisiert)



Ziel

- Die Karte soll für westliche Betrachter durch Verwendung der lateinischen Schrift lesbar gemacht werden.
- Wann immer möglich, sollen lokalisierte Daten aus dem Openstreetmap-Datensatz selbst verwendet werden.
- Andere Lokalisierungstechniken wie Transkription und Transliteration sollen zum Einsatz kommen, wenn ein Objekt selbst keine lokalisierten Daten enthält.

Lösung mit PostgreSQL „stored procedures“

Vorteil:

- Die Lokalisierung ist unabhängig vom Renderer. Es kann jeder Renderer verwendet werden, der PostgreSQL als Datenquelle unterstützt. (Mapnik, Mapserver, Geoserver, ...)

Nachteil:

- Die Lokalisierung steht nur bei Verwendung von PostgreSQL als Datenquelle zur Verfügung. Es gibt keine Möglichkeit der Lokalisierung von anderen Datenquellen wie Shapefiles etc.

Aktuelle Implementierung

- Es stehen folgende PL/pgSQL Funktionen zur Verfügung:

```
osml10n_get_placename
osml10n_get_streetname
osml10n_get_name_without_brackets
```

- Als Zielsprache kann jede Sprache dienen, die das lateinische Schriftsystem verrwendet (Funktionsparameter).
- Die Funktionen können recht einfach hinter Datenbank-Sichten (VIEWS) versteckt werden. Die dadurch entstehenden virtuellen Tabellen erlauben die Weiterverwendung existierender Kartenstile mit minimalen Änderungen (Verwendung einer anderen Namensspalte).

Aktuelle Implementierung

Entscheidungsfindung welcher Name verwendet wird:

- Wenn der Name in der Zielsprache vorhanden ist (bei deutsch z.B. *name:de*), dann diesen verwenden.
- Andernfalls internationalen Namen (*int_name*)
- oder englischen Namen (*name:en*) verwenden.
- Wenn ein lateinisch geschriebener Name vorliegt (*name*), dann diesen verwenden.
- Wenn alle diese Namen fehlen Transkription verwenden.

Transkription und Transliteration

- Transliteration ist die zeichenweise, umkehrbare Umsetzung anderer Alphabete in das Lateinische.
- Transkription soll dem Nicht-Muttersprachler eine halbwegs richtige Aussprache des Wortes ermöglichen und ist nicht notwendigerweise umkehrbar.

⇒ **Transkription ist für unseren Anwendungsfall besser!**

Arten von Schriftsystemen und Transkription

- Alphabete (z.B. lateinisch, griechisch, kyrillisch, arabisch, …)
  - Leicht transkribierbar
- Silbenschriften (z.B. Kana)
  - Relativ leicht transkribierbar
- Logographische Schriftsysteme (z.B. chinesisch)
  - Nur sprachabhängig transkribierbar
- Mischformen (z.B. Thai, Hangeul, …)
  - Relativ leicht transkribierbar

Bekannte Probleme bei der Transkription

- Die Transkription chinesischer Zeichen muss ortsabhängig unterschiedlich erfolgen.
  Da über die PostGIS-Datenbank das Land ermittelt werden kann, in dem sich ein Objekt befindet, konnte diese Lösung für Japan implementiert werden.
- Für das in Thailand übliche Transkriptionssystem „Royal Thai General System of Transcription“ ist mir keine (freie) Bibliothek bekannt. Die in ICU eingebaute Transliteration nach ISO11940 erzeugt unübliche lateinische Schreibweisen.
- In einigen Schriftsystemen (arabisch, hebräisch) werden nicht alle Vokale geschrieben.
  Die Transliteration ist daher in diesem Fall oft unvollständig.
  Beispiel: Die Transliteration von تهران (Teheran) mit ICU liefert „thrạn“

Implementierung der Transkription

- Verwendung der freien Bibliothek *International Components for Unicode* (ICU). Diese stellt eine Transliteration zur Verfügung.
- Es wurde C-Code erstellt, der die Transliteration (*Any-Latin*) aus der ICU-Bibliothek direkt als „stored procedure“ zur Verfügung stellt.
- Ortsabhängige Verwendung einer anderen Transkriptionsbibliothek:
  - Derzeit wird für chinesische Zeichen (im japanischen Kanji genannt) die Bibliothek KAKASI verwendet, wenn sich das zu rendernde Objekt in Japan befindet.
  - Erweiterbar auf andere Schriftsysteme und Länder
- Schriftsystemabhängige Verwendung anderer Transkriptionsbibliotheken kann einfach eingebaut werden.

Verschiedene Schriftzeichen innerhalb einer Beschriftung

Problem:

- Es sind nur wenige Schriftfonts erhältlich, die Zeichen aller Schriftsysteme in der Unicode-Tabelle in gleichbleibend guter Qualität enthalten. Dies ist insbesondere ein Problem, wenn der lokale Name in nicht-lateinischer Schrift in Klammern erscheinen soll, denn die verfügbaren Renderer können die Schriftart oft nicht innerhalb eines Bezeichners ändern.
- Kompromisslösung bei derzeitiger Implementierung:
  Lokaler Name nur bei lateinischen, griechischen und kyrillischen Zeichensätzen in Klammern.

Lösungsansatz:

- Man könnte eventuell einen Mischfont erstellen, der alle relevanten Schriftzeichen in guter Qualität enthält.
- Renderer können so erweitert werden, dass Schriftfonts abhängig vom verwendeten Schriftsystem ausgewählt werden.

Politische Probleme bei der Lokalisierung von Karten

- Viele Regionen auf der Welt waren in der Vergangenheit Teil anderer Staaten. Beispiele: Ehemalige Kolonien, deutsche Siedlungsgebiete in Osteuropa
- So haben beispielsweise selbst kleinste Dörfer im heutigen Polen, im Elsass und in Lothringen einen deutschen Namen. Ob diese heute noch gebräuchlich sind oder nicht, lässt sich oft auch nicht mit Sicherheit sagen.
- Mangels Alternative hoffen wir, dass die Erfasser der Daten nur Namen verwenden, die heute noch gebräuchlich sind.
- Kompromisslösung beim Rendern:
  Darstellung des aktuellen lokalen Namens in Klammern.
  Beispiel: Stettin (Szczecin)

Ausblick und mögliche Verbesserungen

- Technische Lösung des Problems der Darstellung verschiedener Schriftsysteme innerhalb einer Objektbeschriftung.
- Einbau weiterer und/oder besser geeigneter Bibliotheken für die Transkription
- Weitere Differenzierung bei der Transkription nach Ort des Objekts und/oder verwendeten Schriftsystems.
- Abkürzung für „Straße“ in weiteren Sprachen implementieren (derzeit deutsch, englisch und russisch)
- Vorschläge und Anregungen aus dem Publikum insbesondere von Muttersprachlern

Code: https://github.com/giggls/mapnik-german-l10n