

Using shape, proximity and functionality to define neighborhoods with morphologically similar buildings

Raphaël Bubloz

Urban Patterns session
7 September 2019



Unil | Université de Lausanne

Institut de géographie
et durabilité



Introduction

Quick theoretical overview

Urban Morphology

Building shapes

Methodology

Case study

Data used

Process overview

Early results

Conclusions

References

What, how, and why ?

- ▶ Trying to cluster morphologically similar buildings in an urban context
- ▶ Using features derived from simple building representations
- ▶ Clustering using a geo-constrained Self-Organizing Map (geo-SOM)
- ▶ Find (in-)homogeneous places in cities, compare internal morphologies of cities, etc.

Approaches

Literally the *study of urban form*, with several approaches (Kropf, 2017)

- ▶ Typo-morphological (evolution of building types)
- ▶ Configurational (topological relations, influence of spatial configuration)
- ▶ Historico-geographical (townplan study, hierarchy of elements)
- ▶ Spatial analytical (complex systems, physical and socio-economical dynamics)

Similar works

- ▶ Schirmer and Axhausen (2015;2019) proposed a set of features at multiple scales on which to apply clustering. They used it to quantitatively describe the urban morphology
- ▶ Steiniger *et al.* (2008) morphologically characterized building geometries and classified them for cartographic generalization
- ▶ Fan, Zipf, and Fu (2014)

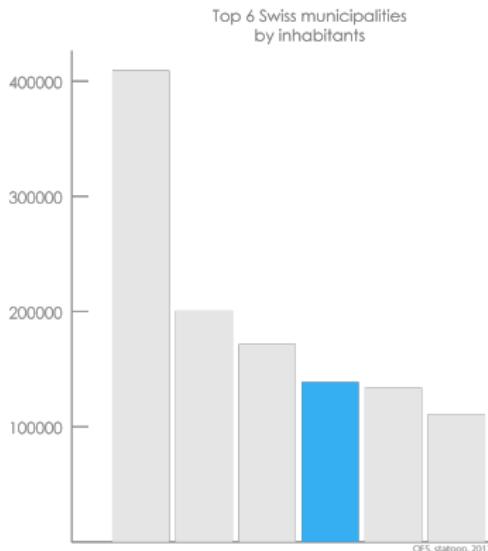
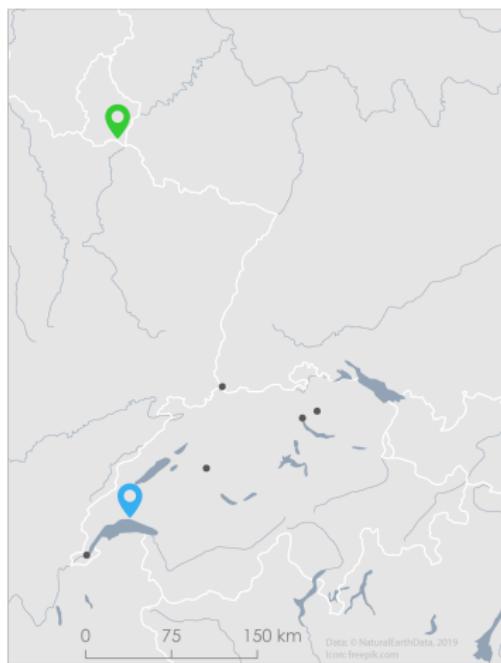
Building shapes

Similar works

- ▶ Schirmer and Axhausen (2015;2019)
- ▶ Steiniger *et al.* (2008)
- ▶ Fan, Zipf, and Fu (2014) used a turning function to find similar footprints of buildings. They tried to classify the morphologies into (functional) types. They also developed a rule-based approach to estimate the type of a building according to its characteristics

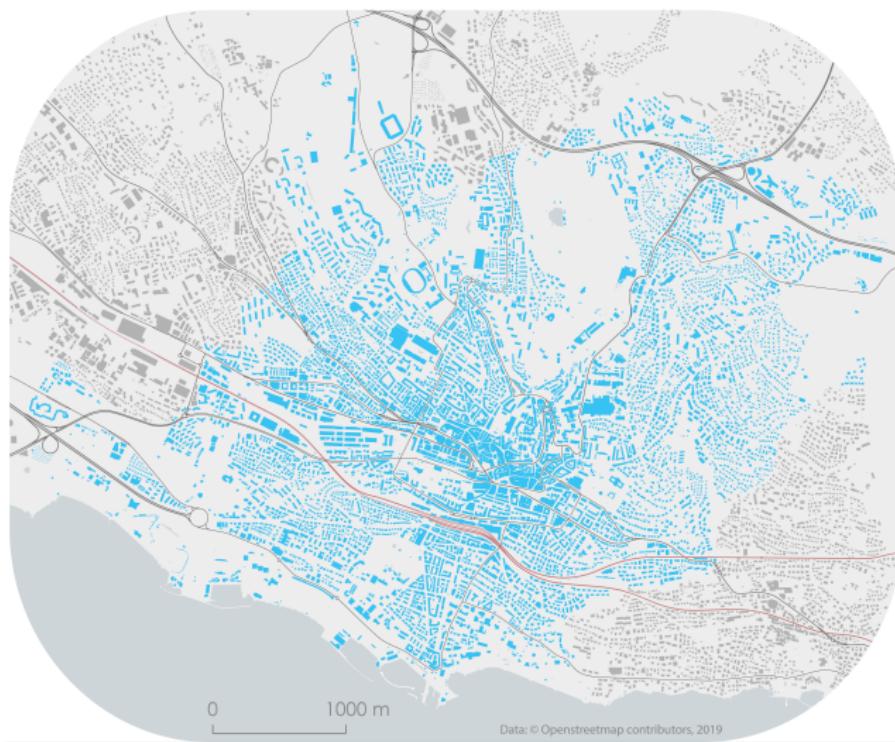
Case study

Lausanne, Switzerland



Case study

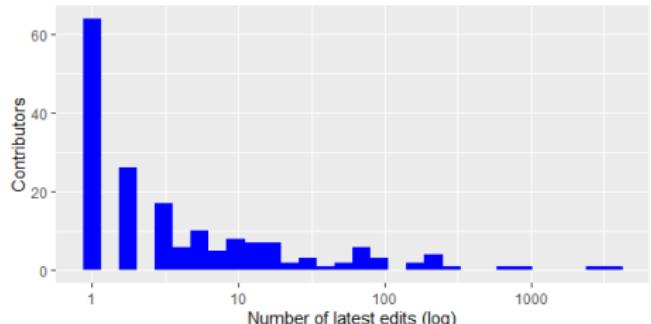
Lausanne Municipality



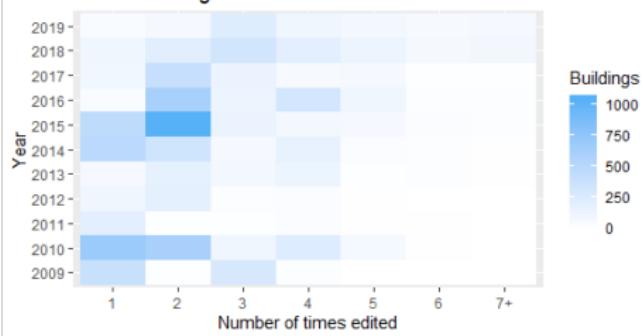
OpenStreetMap data - Gathering footprints

- ▶ Open data generated by multiple contributors
- ▶ Available on a global scale but unevenly distributed
- ▶ Quality depending on several factors (available imagery, familiarity of the cartographer, etc.)

Contributors statistics on buildings



Latest building edition



Selected features overview

OSM buildings in Lausanne	9850	
height specified	3	0.03%
number of levels specified	549	5.57%
type specified	2089	21.21%
levels and type specified	451	4.58%

OpenStreetMap data - Gathering footprints

- ▶ Open data generated by multiple contributors
- ▶ Available on a global scale but unevenly distributed
- ▶ Quality depending on several factors (available imagery, familiarity of the cartographer, etc.)

- ▶ Limits: High geometrical variability, features of interest not sufficiently filled

Data used

Swisstopo data - Computing height

- ▶ Digital Elevation Model, 2016, 2m resolution, 0.5m accuracy
- ▶ Digital Surface Model, 2016, 2m resolution, 0.5m accuracy
- ▶ Compute the difference and assign mean value of cells intersecting each building
- ▶ Gives a sufficient approximation of the building height

Data used

Buildings and housing registry - Retrieving categories

- ▶ Central registry maintained by the Federal Statistical Office
- ▶ Data input by the municipalities
- ▶ General categories (6) and specific classes (26) adapted from EUROSTAT
- ▶ Quality assurance ?

Indicators derived - Feature creation

- ▶ Merged adjacent buildings and remove odd valued ones (e.g. negative height)
- ▶ Features based on the footprint geometry and the height
- ▶ Inspired by Schirmer and Axhausen (2015) with several additions (oriented envelope, compactness)
- ▶ Quite correlated and redundant
- ▶ How to select the most suitable ones

Model-based clustering

- ▶ Using the method implemented by Ceuleux *et al.* (2014)
- ▶ Simultaneously selects the number of clusters and a set of *relevant* features
- ▶ Gives a clustering based on features (morphologic typology)

(Geo-)SOM

The Self-Organizing Map was proposed by Kohonen (2001)

- ▶ From a high-dimensional dataset to a simpler (2D) output/representation
- ▶ Initialisation of neurons with a weights vector corresponding to the dimensions of the dataset
- ▶ Competitive and adaptive learning

Geo-SOM

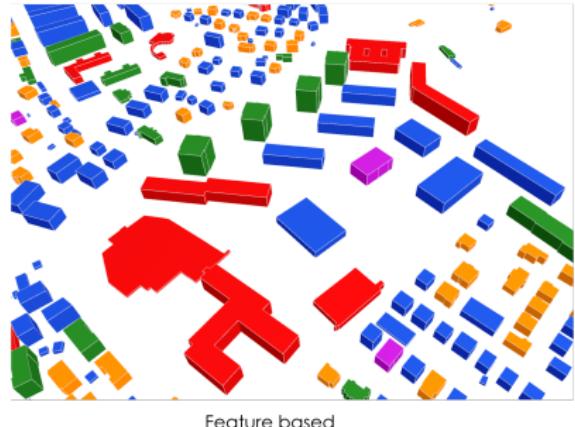
A SOM with a geo constraint Baçao *et al.* (2005)

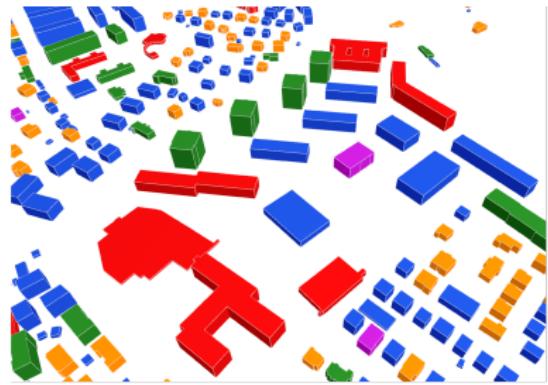
- ▶ Take the data point
- ▶ Find the (spatially) closest neuron
- ▶ Look into it and its neighbours to find the most similar
- ▶ Adjust the BMU and its neighbours and repeat with the next data point until set number of iterations
- ▶ At the end, apply a K-Means on the neurons
- ▶ Different adaptable parameters: size of the map, neighbourhood type, number of searchable neighbours

Input

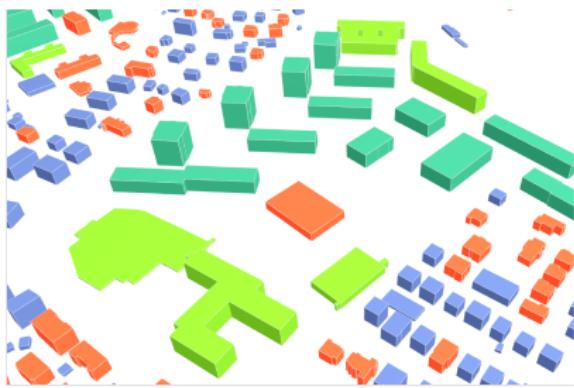
The following results examples are based on an input of 15 hand-picked features

For this input, the GMM selects 10 features and 5 clusters

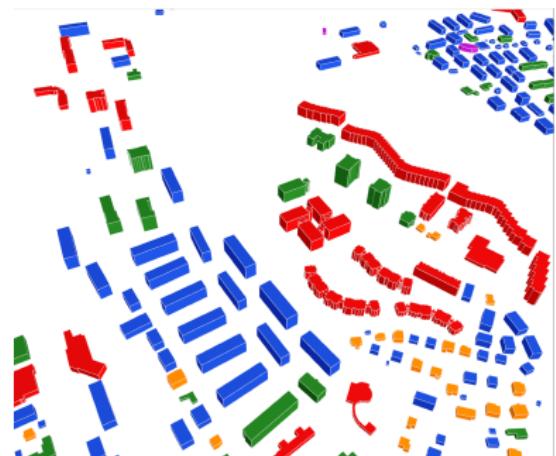




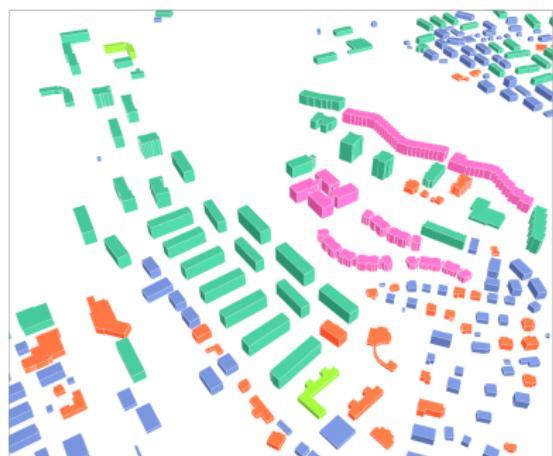
Feature based



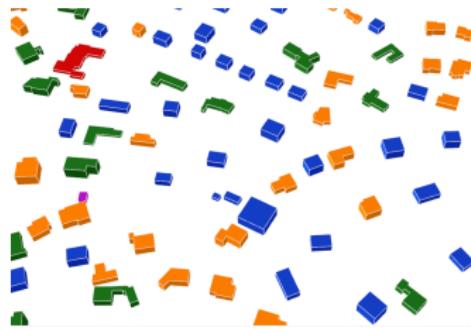
Geosom based



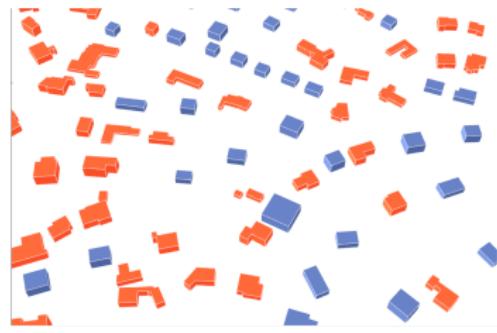
Feature based



Geosom based



Feature based

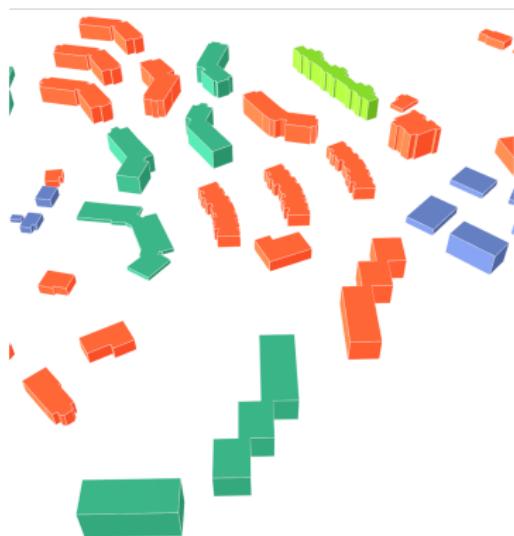


Geosom based

Still progress to be made



Feature based



Geosom based

Main difficulties and open questions

- ▶ Clustering and lack of results comparisons
- ▶ Garbage in, garbage out
- ▶ Can the features allow the discovery of relevant results ?
- ▶ *What is a building ?* (garages, sheds, stadiums, etc.)

Future works

- ▶ Experimenting with the geo-SOM parameters
- ▶ Switch from the centroids to a proximity-based relation
- ▶ Incorporate the functions of buildings in a meaningful way
- ▶ Try an approach based on computer vision

Thanks for your attention

Slides, routines (and more) will be available on *GitHub*

<https://github.com/Raphbub/>

For any further questions, remarks or suggestions
raphael.bubloz@unil.ch



Bação, F., Lobo, V., & Painho, M. (2005). The self-organizing map, the Geo-SOM, and relevant variants for geosciences. *Computers & Geosciences*, 31(2), 155-163.

Celeux, G., Maugis-Rabasseau, C., & Sedki, M. (2019). Variable selection in model-based clustering and discriminant analysis with a regularization approach. *Advances in Data Analysis and Classification*, 13(1), 259-278.

Fan, H., Zipf, A., & Fu, Q. (2014). Estimation of Building Types on OpenStreetMap Based on Urban Morphology Analysis. In J. Huerta, S. Schade, & C. Granell (Éd.), *Connecting a Digital Europe Through Location and Place* (p. 19-35).

Kohonen, T. (2001). *Self-organizing maps* (3rd ed). Berlin; New York: Springer.

Schirmer, Patrick M., & Axhausen, K. W. (2019). A Multiscale Clustering of the Urban Morphology for Use in Quantitative Models. In L. D'Acci (Ed.), *The Mathematics of Urban Morphology* (p. 355-382).

Schirmer, Patrick Michael, & Axhausen, K. W. (2015). A multiscale classification of urban morphology. *Journal of Transport and Land Use* 9(1).

Steiniger, S., Lange, T., Burghardt, D., & Weibel, R. (2008). An Approach for the Classification of Urban Building Structures Based on Discriminant Analysis Techniques. *Transactions in GIS*, 12(1), 31-59.

Geo-SOM

A Self-Organizing Map with a geographical constraint

Data normalized into range $[0, 1]$

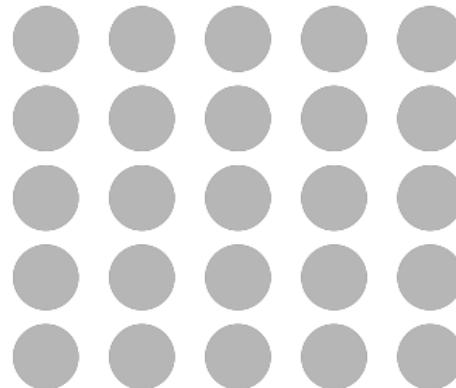
Data as n -dimensional vectors d

$$d = [x_{coord}, y_{coord}, \xi_1, \dots, \xi_n]^T$$

Initialisation of a map with

$l \times m$ neurons n

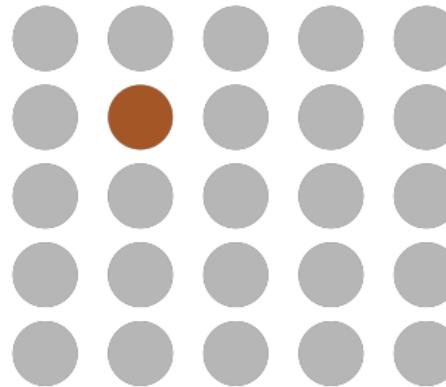
$$n = [x_{coord}, y_{coord}, \mu_1, \dots, \mu_n]^T$$



Geo-SOM

For each d , find the most
spatially similar neuron

$$Geo_{BM} = \min_i ||d_{xy} - n_{xy}||$$



Geo-SOM

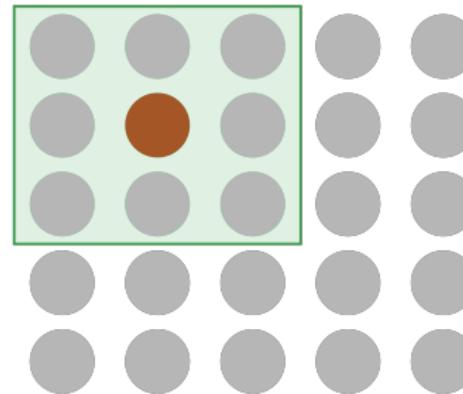
For each d , find the most spatially similar neuron

$$Geo_{BM} = \min_i ||d_{xy} - n_{xy}||$$

Then, find the most similar neuron in its neighbours

$$BMU = \min_j ||d - n||$$

with $n \in Geo_{BM}$ neighbours



Geo-SOM

The BMU and its neighbours
are updated

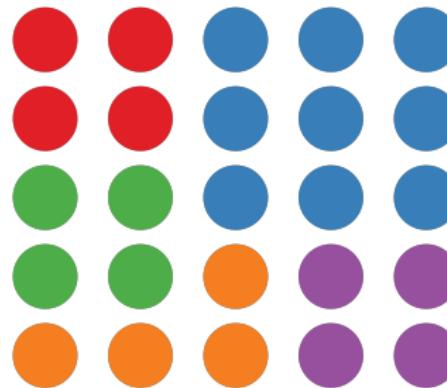
$$n_i(t+1) = n_i(t) + h_{ci}(t)[d(t) - n_i(t)]$$

Geo-SOM

The BMU and its neighbours
are updated

$$n_i(t+1) = n_i(t) + h_{ci}(t)[d(t) - n_i(t)]$$

K-Means is applied on the
final SOM (Flexer, 2001)



Geo-SOM

The BMU and its neighbours
are updated

$$n_i(t+1) = n_i(t) + h_{ci}(t)[d(t) - n_i(t)]$$

K-Means is applied on the
final SOM (Flexer, 2001)

Data vectors are assigned
groups using the previous steps

