

A platform for spatial data labelling in an urban context

Julien Lesbegueries, Nicolas Lachiche, Agnès Braud, Grzegorz Skupinski, Anne Puissant, Julien Perret

▶ **To cite this version:**

Julien Lesbegueries, Nicolas Lachiche, Agnès Braud, Grzegorz Skupinski, Anne Puissant, et al.. A platform for spatial data labelling in an urban context. International Opensource Geospatial Research Symposium (OGRS 2009), Jul 2009, Nantes, France. 11 p. halshs-00626859

HAL Id: halshs-00626859
https://shs.hal.science/halshs-00626859

Submitted on 16 Jun 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A platform for spatial data labeling in an urban context

Julien Lesbegueries[1], Nicolas Lachiche[1], Agnès Braud[1], Grzegorz Skupinski[1], Anne Puissant[1], Julien Perret[2]

(1) Université de Strasbourg, Strasbourg, France. (2) Institut Géographique National, Saint-Mandé, France

1. Introduction

The GeOpenSim project aims at developing an open source framework to study urban evolutions using vector based topographic databases. This framework, based on GeOxygene [1], is composed of several modules spanning from the creation of spatio temporal topographic databases to the simulation of urban dynamics. The novelty of our approach lies in using topographic data for the simulation whereas most research on urban simulation is based on cellular automata or graph cellular automata [2, 3, 4, 8, 9]. Indeed, the analysis of observed urban phenomena using topographic data allows for a more accurate and more realistic approach to urban simulation. Such an analysis is realized at different geographic levels: at the micro level (buildings, roads, etc.) and at several meso geographic levels (urban blocks, districts, cities, etc.). Those levels have already been used in other works, e.g. [7, 10]. For each of these levels, the context in which the evolutions take place is crucial to the study. The context of an evolution is time-based (evolutions are different in the 1950s and in the 1970s) as well as spatially based (evolutions in a peri-urban context are different from the evolutions in an industrial context). Therefore, in the framework of this spatial context, geographic features have to be characterized so they can be labeled depending on their nature. This paper presents our work on the la-

beling of a specific type of geographic features: elementary areas (or urban blocks). Here labeling refers to the manual assignment of existing class labels to elementary areas chosen by a human expert. A machine learning technique could then use those examples to extract a model to predict the class labels for remaining elementary areas. The proposed labeling platform could therefore be used to automate the classification of urban areas for eg. sustainable development or urban planning.

We developed an open source add-on to Geoxygene. This labeling add-on fulfills several needs concerning the management of geographic data visualization and labeling. Its input consists in geographic data and a list of labels (screenshot in figure 1). Geographic data consists in several layers of topographic data, at micro and meso levels (figure 1 (A)). A target layer is selected in order to define the elements to be labeled (for example, areas in figure 1(C)). The other ones are used in the visualization. Moreover a list of labels is defined (figure 1 (D)) and the module generates a dynamic interface with items or sliders for each label. Sliders allow the user to assign several classes to a single area. If needed, the user can also add a comment (figure 1(B)).

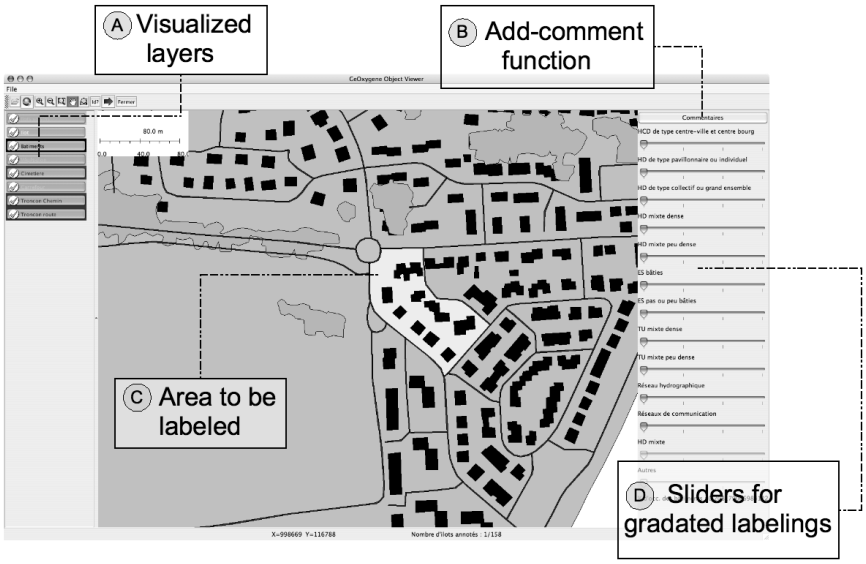


Fig. 1 Global screenshot of the labeling module

The module provides a labeling process with several options in order to perform various experiments and to fit to different problems:

- several visualizations are provided,

- several ways of labeling,
- several procedures to complete the labeling.

The different options are explained and illustrated with the specific problem of urban labeling.

The figure 2 summarizes the main procedure of labeling. A cartographic window is created from a geographic database with connection functions provided by Geoxygene [1] for specific geographic data. A user logs in to the interface and labels selected areas manually. The labels are stored in a database and export functions produce appropriate output files or output storing in training databases. This data is then used to perform a learning process on the entire dataset. One way is to integrate learning functionalities in our platform with the help of the Open Source Data Mining toolkit Weka [12]. This allows us to analyze learning results directly in the platform.

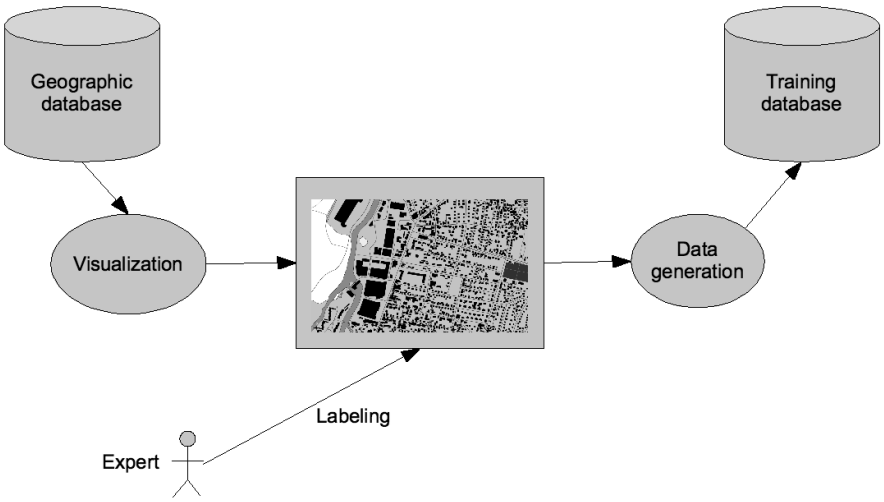


Fig. 2 Labeling procedure schema

In the next section, facilities concerning the problem modeling are described: firstly how to carry out the labeling process, and secondly different ways of labeling. Section 3 presents the labeling acquisition, its analysis and its use for a supervised learning purpose.

2. Modeling the problem

This section describes how the platform helps modeling the problem. A first step consists in defining the geographic input data and once the geographic parcels are defined, the second step consists in defining the labels.

In the framework of our problem, we must define adequate geographic parcels in a city in order to locate potential evolutions. The first step is to partition it in relevant elementary areas (districts). This partitioning must be generic and automatic. Then labels must be defined to categorize these areas according to their evolution potential. The former function of urban partitioning is performed within Geoxygene and the labeling module performs the latter.

2.1. Input data

Input data must be geographic layers stored in a geographic database e.g. PostGIS[1]. One of the layers must be the target one, in which geographic parcels have to be labeled.

To provide a generic labeling module, we chose to use the topographic database, provided by the French National Geographic Institute (IGN). Indeed there is a vector based topographic database, namely the BDTopo, available for the entire French territory and thus for every French city. This database is composed of layers for each kind of *micro* geographic objects (buildings, vegetation, networks, etc.).

The partitioning of the city is computed according to the communication network layers: main roads, country roads, railroads, and watercourses. Figure 3 shows the resulting elementary areas that correspond to the spatial unit to label (the light grey one for example).

2.2. Labels definition

The platform allows to dynamically define labels, because they can change all along the labeling campaign. A property file configured *a priori* is used to create a label index in a database (figure 4) and to build the list of sliders (figure 1 (C)).

We illustrate this step on our urban problem. Labels are similar to those used in previous studies such as [11]:

[1] http://postgis.refractions.net/

1. Continuous urban fabric (city center),
2. Discontinuous urban fabric with individual houses,
3. Discontinuous urban fabric with collective buildings,
4. High density mixed housing surface (mix of 2 and 3),
5. Low density mixed housing surface (mix of 2 and 3),
6. Specific urban surface (industry buildings),
7. Not or little built specific urban surface (industrial wasteland),
8. High density mixed urban surface (mix of 2,3 and 6),
9. Low density mixed urban surface (mix of 2,3 and 6),
10. Hydrographic network (canals, rivers),
11. Communication network (roads, country roads, railroads).

For each label, a slider and a tuple in the database are created. This dynamic building is necessary because the labeling procedure is intrinsically an iterative, possibly backtracking procedure. Next section details this characteristic.

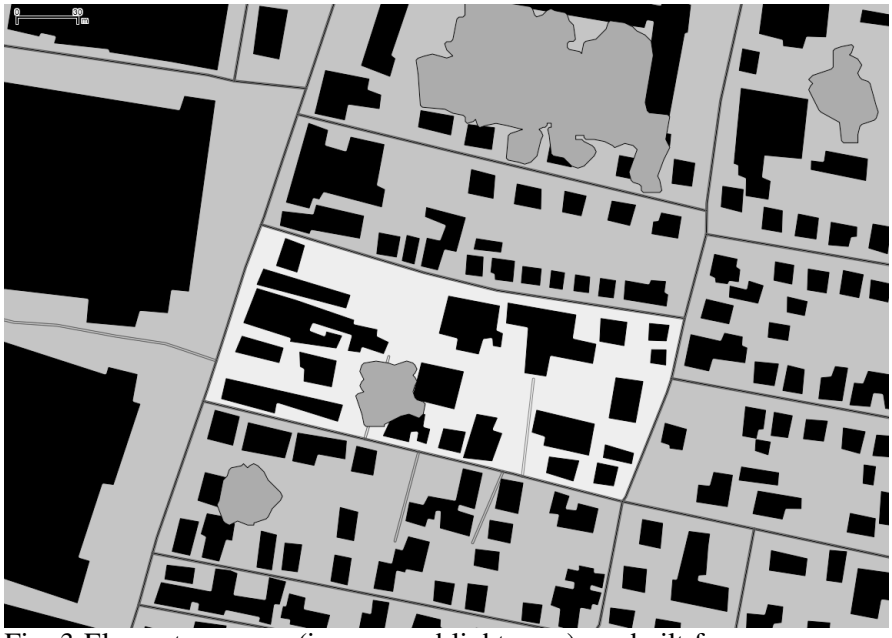


Fig. 3 Elementary areas (in grey and light grey) are built from communication networks (lines). The black polygons represent buildings and the rounded areas represent vegetation.

2.3. Refining the labels

The module provides different functionalities to refine the labeling.
A relevant question is the manner labels are collected.

The easiest way is to force the expert using the platform to choose one class label for each visualized area during the labeling process. However, in the case of a random choice of next areas to be labeled, the expert may have difficulties to label every area, given that the areas generation is automatic and can produce artifacts. An additional class "others" can solve the problem.

A second implemented solution consists in allowing a gradation the expert can express with a confidence degree from 1 to 4. In this case the value of each class is stored for each labeled area.

Another manner to face the difficulty of labeling is to consider areas labels as overlapping classes (the area is no longer "of one kind" but "made of"). Then the expert is allowed to associate several classes by area (with gradations for each class).

During our labeling campaign, some mixed classes were added in order to disambiguate confusing areas (4, 5 and 8, 9).

2.4. Confidence level and exclusive / overlapping classes

This section describes the various labeling functionalities implementation. The user can label in a binary manner or in a more gradated manner. The binary manner can be sufficient for easy-to-label procedures, when human experts do not have any doubt. When the labeling procedure consists in a more complex problem, implying confusing areas to label, the gradated manner implemented by sliders and representing a confidence degree can be a solution. By default, the module expects 4 gradations (from 0 to 3). The figure 4 presents a schema of the labeling database (storing binary and gradated labelings). The *labeling index* table stores the expert identifier, the concerned area identifier and its labeling. In the binary way, the *label_id* column is used whereas in the gradated way it is the *gradated_label_id* one (on the figure, both columns are filled in order to illustrate the 2 cases). Then the *Gradated index* stores the sliders values.

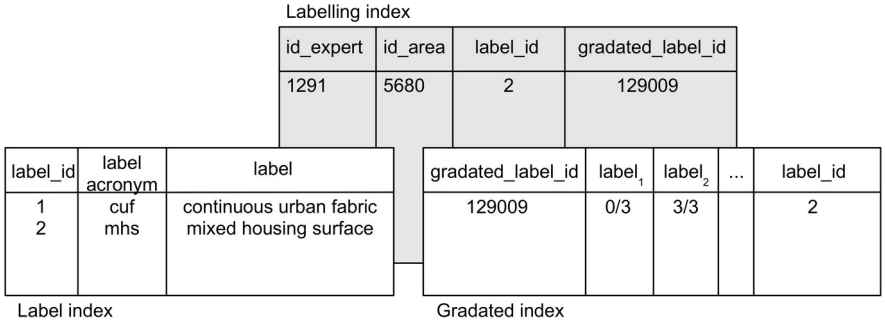


Fig. 4 Schema of the labeling database

Moreover, the labeling process can be a one label per area labeling or a several labels per area labeling. It depends on the labels definition. Indeed, they can represent exclusive classes or overlapping classes. In our case study, labels cannot be exclusive because an area can contain *housing surface* and *specific urban surface* at a time. Our problem turns out to be an exclusive labels case, if an additional specific one (*mixed urban surface*) is defined. However the labeling structure can manage the two solutions. The only change is the function used to determine the *label_id* column of the gradated index (max of the $label_i$ columns for instance).

2.5. Identifying the minimum background

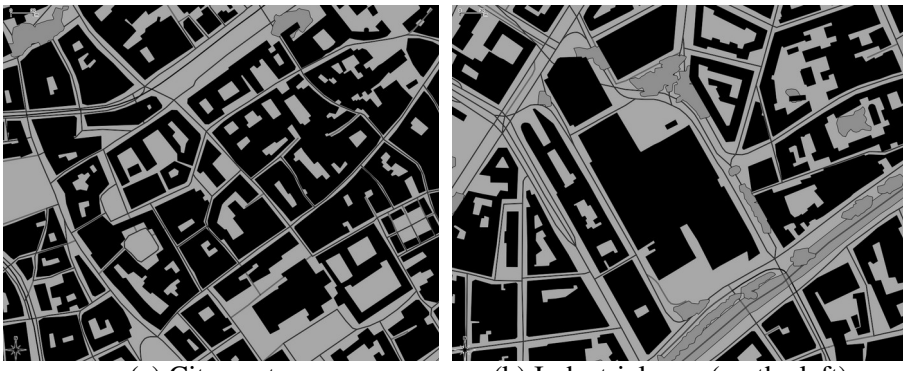
Additionally, the module provides different manners to choose areas to be labeled. Indeed, in a learning perspective, areas have to be well chosen in order to correspond to a representative set. Several solutions are provided (randomly-based, user-based, active learning).

Once input data and classes are correctly defined, we experiment several ways of labeling in order to find the best method requiring the minimum information to display, providing however sufficient information to experts and identifying the adequate retrieval of information necessary for the learning task.

Three visualizations are proposed in our platform (figure 6), providing different widths of spatial context:

1. the area to label only (with its buildings),
2. the area to label and its surrounding areas (with their micro objects),
3. the entire map of the city.

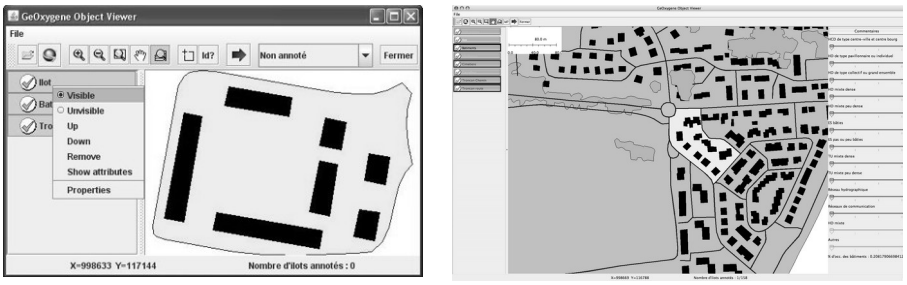
In our case study, experts claim that the first visualization is too poor and there is not enough information to make a decision. For example, it is difficult to distinguish a city center area from a specific urban one (industrial area for instance) without the direct neighborhood and their scale (figure 5).



(a) City center (b) Industrial area (on the left)

Fig 5. Confusing areas difficult to label

The two other visualizations provide this additional contextual information. However, there is a problem of *over-contextualization* for the third one because the visualization of the entire map causes the expert to use his background knowledge of the geographic area (experts could then recognize the displayed city) instead of the conformation of objects in areas.



(a) (b)



(c)

Fig 6. Three visualization widths offered by the module

3. Data processing

This section presents module facilities concerning the automation of the labeling procedure: data acquisition, analysis and visualization, and its use in a learning process.

3.1. Data Acquisition

The problem depicted here concerns the manner areas to label are chosen. Several solutions are proposed on the platform. One solution is to visualize a random area not yet labeled (figure 6(a) and 6(b)). An alternative solution provided consists in allowing the expert to choose the areas to label (figure 6(c)).

The aim of the platform is to include a learning process, able to label the majority of areas from a few examples. With the second solution, experts have to carefully choose the areas. They should not select the ones easy to label only. If the target of the learning process is to label every area, it must be trained from typical and also fuzzier / tricky / complex areas. Another solution would be to use active learning methods [5] to select next areas to label.

3.2. Analysis and visualization of labelings

This section presents additional functionalities in order to compare labeling between experts. Indeed, it is important to check whether different experts agree on the chosen labeling process: the set of labels and their definition, and the sample of examples.

A particular attention is given to the agreement of labeling between them. First of all, a visualization of the labeling progress is provided for each expert, for each part of the city, in order to indicate to label in priority the areas that have already been labeled by other experts. Then an agreement visualization (figure 7) allows seeing which areas are the most conflicting ones. The agreement measure is (Eq. 1).

$$U = \frac{\sum_{i=1}^{p} \sqrt{\sum_{c=1}^{n} (m_c - e_i(c))^2}}{p} \quad (1)$$

where

- e_i denotes the experts (there are p experts),
- c denotes the classes (the cardinality of the set of classes is n),
- m_c denotes the average over the different experts of the values for the c class,
- $e_i(c)$ denotes the value chosen by the ith expert for the c class

A greater value of U denotes a stronger disagreement of the experts.



(a) Agreement of experts with first settings | (b) Agreement of experts with other settings | (c) Legend

Fig. 7 Agreement variation along the experiments

The figure 7 shows a clear improvement between the first and the latter labeling process. These kinds of analysis allow the experts to choose the best way of labeling.

In order to eliminate residual conflicts, we created a functionality allowing experts to add comments (figure 1(B)) to their labeling and to visualize areas by comment (not-well formed area, city border area, etc.).

3.3. Towards automating the labeling process

Finally, export functionalities allow using labeling results in learning programs. In particular, exports in arff format for Weka [12], an Open Source learning toolkit, and in 1BC format for 1BC, an ILP[2] learning tool [6] are provided. This is a first step to automate the labeling from a training set. The figure 8 shows an excerpt of automatically labeled map, produced by a machine learning algorithm applied on experts training sets from the urban labeling procedure.

A part of the Weka library will be integrated in the module in order to envisage a semi-automatic learning process based on a supervised classification. Moreover, this integration allows us to imagine active learning functionalities [5] that could improve the labeling by choosing the most appropriate areas to label for an efficient learning process.

[2] Inductive Logic Programming



Fig. 8 Predictions of a supervised classifier trained from data labeled thanks to the labeling platform

4. Conclusion

This paper presents the manual labeling functionalities of a Geoxygene[3] extension, within the framework of a urban classification. The module aims at being generic in order to perform similar classifications for other geographic layers. We investigated pertinent facilities for a labeling module, like the labels management, the confidence level capability, the exclusive or overlapping classes choice, the visualizations, and the analysis and use of labeling results.

Future works concern experiments of the integrated learning module, in order to classify the whole dataset and the active learning option allowing

[3] GeOxygene aims at providing an open framework which implements OGC/ ISO specifications for the development and deployment of geographic (GIS) applications. It is a open source contribution of the COGIT laboratory at the IGN (Institut Géographique National), the French National Mapping Agency. It is released under the terms of the LGPL (GNU Lesser General Public License) license.

the machine to choose the best training examples, i.e. to speed up the collection of training data.

References

1. Badard T, Braun A (2003) Oxygene – d'une plate-forme interopérable au déploiement de services web géographiques. Revue internationale de géomatique 3(13):411-430
2. Barros JX (2003) Simulating urban dynamics in latin american cities. In proceedings of the 7th international conference on GeoComputation, University of Southampton, United Kingdom
3. Batty M (2005) Cities and complexity: understanding cities with cellula automata, agent-based models, and fractals. MIT Press
4. Benenson I, Portugali J (1997) Agent-based simulations of a city dynamics in a gis environment. In COSIT '97: Proceedings of the International Conference on Spatial Information Theory, pp 501-502, London, United Kingdom, Springer-Verlag
5. Bondu A, Lemaire V (2007) État de l'art sur les méthodes statistiques d'apprentissage actif. RNTI, Numéro spécial dur l'apprentissage et la fouille de données
6. Flach P, Lachiche N (2004) Naive Bayesian Classification of Structured Data. Machine learning 57(3):233-269
7. Gaffuri J, Trévisan J (2004) Role of urban patterns for building generalisation: an application of AGENT, 7th ICA Workshop on Generalisation and Multiple Representation, 20-21 august, Leicester (UK)
8. Hammam Y, Moore A, Whigham PA (2007) The dynamic geometry of geographical vector agents. Computers, Environment and Urban Systems 31(5):502-519
9. O'Sullivan D (2001) Graph-cellular automata: a generalised discrete urban and regional model. Environment and Planning B: Planning and Design 28:687-705
10. Steiniger S, Weibel R (2007) Relations among map objects in cartographic generalization. Cartography and Geographic Information Science (CaGIS) 34(3): 175-197
11. 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

12. Witten IH, Frank E (2005) Data mining: practical machine learning tools and techniques (second edition). Morgan Kaufmann