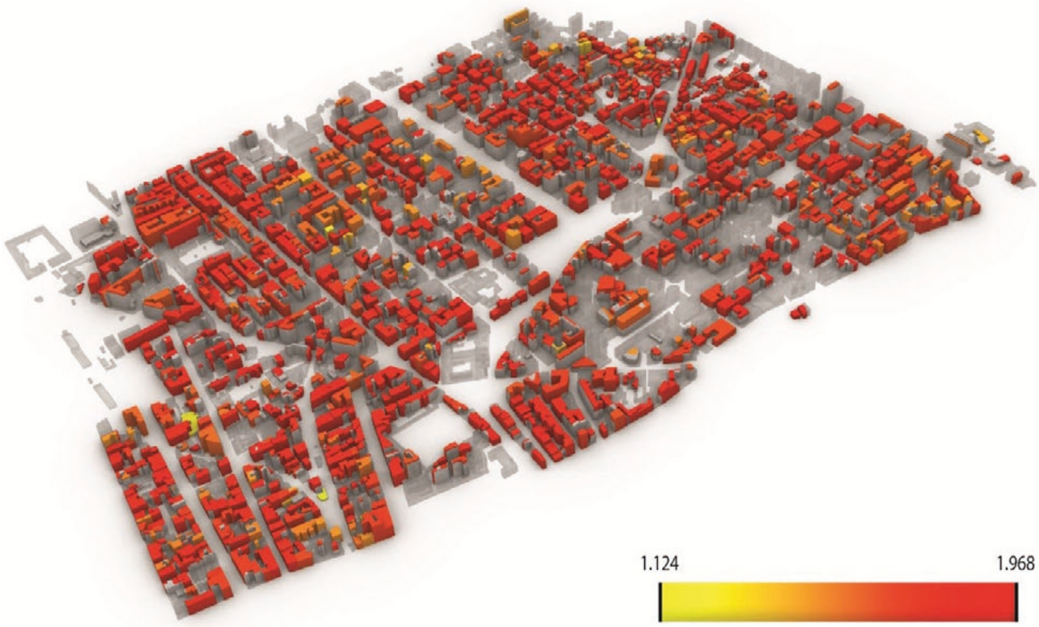
AI for AEC: Open Data and VPL Approach for Urban Seismic Vulnerability

Federico Mario La Russa

Abstract

This contribution provides an overview of the VPL evolution and an application case concerning the classification of seismic vulnerability indices with AI. This research aims to contribute to the scientific debate on the use of these technologies in architecture, deepening the themes of seismic assessment on urban and territorial scale. The whole experimentation was conducted using only the potential of Grasshopper's VPL and possessing, as basic knowledge, the main concepts of machine learning and supervised learning. The VPL is therefore an effective tool to introduce and disseminate the topics and applications of artificial intelligence within the AEC sector, effectively decreasing the gap between domain experts and programmers.

Keywords
open data, VPL, AI, seismic assessment, CIM – city information modeling.



doi.org/10.3280/oa-686.61

Introduction

Over the past few decades, the human–machine relationship has progressed significantly due to the evolution and deployment of increasingly advanced technologies. Among all of them, Artificial Intelligence (AI) has become prevalent in several application fields, including Architecture and Construction (AEC) industry. One of the main objectives in the AEC sector is to develop semi–automated solutions and workflows that can minimize repetitive and time–consuming activities, thus allowing professionals to focus on more valuable and relevant tasks. In this direction, progress had already been made through the widespread adoption of Building Information Modeling (architectural scale) and City Information Modeling (urban and territorial scale). These digital ecosystems usually do not natively possess tools and/or interfaces that allow professionals to apply AI to their models. Thus there are few applications in the field and mainly developed in the academic world where it is easier (for a generic domain expert) to develop computational skills and interface with other programmers. The gap between 'designer' and 'programmer' has been reduced with the introduction of Visual Program Languages (VPLs) within modeling software to develop computational codes. Their ease of use lies in their visual nature and in a vocabulary of 'components' where the main grammatical rule consists in the relationship between input and output.
The research aims to investigate what role Artificial Intelligence can play in the urban survey and City Information Modeling for the mapping of seismic vulnerability. For this purpose, the following research questions have been defined:

- Is it possible through a VPL to determine the relationship that links characteristics of building units with their corresponding seismic vulnerability?
- What are the limitations and potential in using a VPL like Grasshopper (GH) for Artificial Intelligence applications?

VPL Evolution and Impact in AEC Industry: History and Reflections

In the 1980s, there was a great diffusion of personal computers, but the average user did not have programming knowledge and this limited the impact of these technologies in different sectors. Programmers tried to improve the user interface but not always the efforts in this direction were successful. This condition led to researches aimed at using graphics to facilitate programming skills, leading to the birth of Visual Programming (VP) [Halbert 1984]. By eliminating syntax, the graphical method focused on workflow, making visual programming an efficient tool even for skilled programmers. The friendliness of this method was also demonstrated by cognitive psychology, as the human brain can process visual information using two hemispheres instead of one as in other cognitive processes [Myers 1986]. In accordance with Brad Myers, VPL can be defined as a "system that allows the user to specify a program in a two (or more) dimensional fashion. Conventional textual languages are not considered two dimensional since the compiler or interpreter processes it as a long, one–dimensional stream" [Myers 1986]. The first VPLs for geometry modeling purposes can be found in the late 80's: Prismis (nowadays known as Houdini) and ConMan [Haeberli 1988]. In the 2000s there was a new success of parametric design with a subsequent spread of programming tools (ex. GH, Dynamo, Marionette) for design purposes. The applications went far beyond that, as the new VPLs allowed the management of entire workflows (and data). VPLs for architecture began to be recognised as programming languages capable of facilitating operations that designers, engineers and architects used to carry out manually [Rutten 2012]. Together with the BIM revolution, these topics started to be included in the training of young architects [Boeykens et al. 2009].
Compared to traditional programming, visual programming has a very favourable learning curve in the short term. However, for more complex processes, VPLs are limited because they cannot keep up with traditional programming in the long term [Zwierzycki 2017].
Thanks to the community behind VPLs such as GH, it is possible to use a series of plug–ins that increase the potential of VPLs compared to their default setup. However, there is still a gap in the long term, even if it is smaller than the previous one. In recent years, there has been an increasing amount of applications in AEC regarding the use of artificial intelligence. A variety of plug–ins have been created that allow the transition to these new practices

within VPLs by reducing the knowledge required to apply them. These plug–ins enable the user to use Machine Learning and Deep Learning tools, enabling increasingly complex data processing practices. Applications range from design to optimisation in production processes. Although in some applications there is no need for textual programming implementations, VPL shows limitations in the long term.

Urban Seismic Risk Assessment: the Italian Methods

In relation to seismic vulnerability assessment at the urban scale, three majors schools of thought have been identified that aim to combine expeditious surveying with accuracy in assessment. These methods differ mainly in the type of data required and the accuracy of the analysis. In particular, there is an inversely proportional relationship between analysis extension and accuracy assessment (the more accurate the assessment, the closer to the architectural scale). Statistical evaluations focus on the determination of vulnerability with reference to different main characteristics of the buildings in order to analyse their distribution over the territory. Other analyses follow a mechanistic procedure where the structural behaviour is studied by simulating seismic actions on the building unit. As regards holistic analysis, it generally begins with an investigation of the urban growth of the fabric in which the building under analysis is located, then it recognises the construction components, maps the decays and analyses instabilities [Corradi et al. 2014; Caliò et al. 2018].

Methods and Workflow

Starting from existing studies on seismic vulnerability, the objective is to classify, through Artificial Intelligence mechanisms, the vulnerability of single building units using a few parameters easily obtainable from qualitative visual surveys. This approach is based on the assumption that each building unit is a living organism with its own genetic code made by all its parameters. In literature we find similar approaches at the architectural scale of BIM models [Tono 2018]. Therefore, it is essential to use programming tools that allow sufficient data granularity for their treatment from the territorial to the architectural scale and the use of Artificial Intelligence tools. The use of VPL based on CAD environment allows to easily interrogate the geometries obtained from the initial data, as well as to visualize the results of the analyses through thematic three–dimensional maps. In the specific case of seismic vulnerability at urban scale, the use of VPL as modelling and analysis tool facilitates the AEC sector professionals in the design phase thanks to the available plug–ins.
The research presents a workflow, developed with only the tools of visual programming (GH), to train a neural network using a dataset of seismic evaluations at the urban scale (statistical method). This approach belongs to supervised learning methods. In particular, a simple validation will be performed by means of a linear regression with several variables that identifies the relationship between indices and vulnerability values thus allowing the prediction of vulnerability in other urban blocks. The workflow can be summarised as follows:

- Downloading and importing the initial dataset;
- Data processing according to the Simple Validation scheme;
- Neural network training using the 'Dodo' plug–in;
- Model validation (coefficient of determination R2);
- Representation of the obtained predictions.

Case Study: the Historical Center of Palermo

To validate the proposed methodology, a dataset with the necessary characteristics was identified. The dataset comes from an open data work developed by the PalermoHub mappers community [1]. In this work, a seismic vulnerability analysis based on statistical methods was made for more than 1500 building units in two areas of the historical centre of Palermo. The whole dataset was created exclusively on open data available online from different institutions such as ISTAT and the Municipality of Palermo [Vitrano 2017].

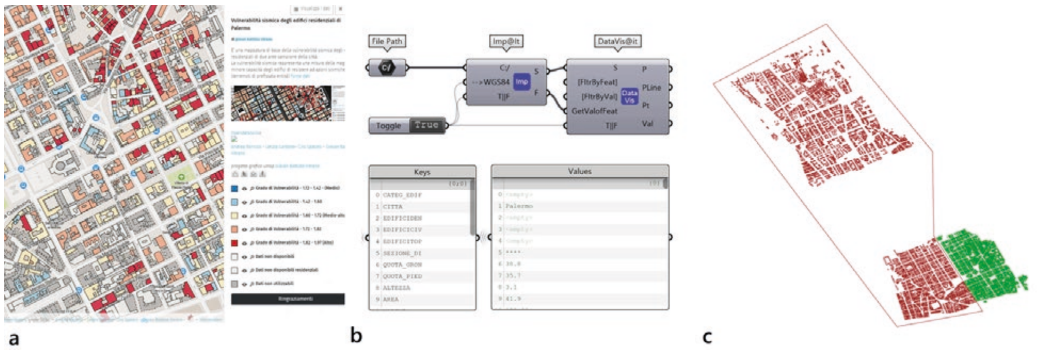


Fig. 1.
a) Initial dataset (webgis) of Palermo historical center;
b) VPL code to import geodata inside Grasshopper;
c) Training dataset (in red) and test dataset (in green).

The dataset is made of indices related to the period of construction, number of floors, construction material, state of preservation and the vulnerability of the building units.
The file (available for online viewing) was downloaded as geojson and converted to a shapefile using QGIS. The conversion into a shapefile enabled the import into GH via the 'at–it' plug–in. A code was then developed with the aim to filter the indices and vulnerabilities of each individual building unit contained in the input shapefile.
In the field of supervised learning, the method of the simple validation requires that the dataset is divided into two parts. The dataset destined to the training of the system usually constitutes 70-90%, the remaining part (30-10%) is destined to validate the training of the model of machine learning. In this case, the part assigned to training is 71.3% (DS1), the part assigned to validation (testing) is 28.7% (DS2). Within these two subsets, a filter was developed (via VPL) to separate the indices of building unit characteristics from the corresponding seismic vulnerabilities (fig. 1).
The open source Dodo plug–in [1] was used to train the neural network. Dodo allows to specify some significant parameters regarding the training process. In particular, the component 'Supervised Training NN' was used for the training of the neural network (it represents the phase of supervised learning). As input, the indices and the relative vulnerabilities of each building unit within the DS1 were given. The output was the neural network structure trained to identify the relationship between indices and vulnerabilities. Then 'Run Neural Network' component was launched giving as input the trained neural network and DS2 indices (test) to predict the vulnerability of DS2 building units.

Results and Discussion

The results of the predictions were used to create a representation of the buildings of the tested set DS2 (in blue) in comparison with the initial values (in grey) (fig. 1). Visual analysis of this 3D map suggested that the neural network identified the relationship between indices and vulnerability. An analytical verification was then made using the coefficient of determination (both standard and reduced) which confirmed this result, returning a value of 0.99 out 1.00. By graphically analysing data distribution, the neural networks managed to fit the test data almost perfectly (fig. 2).
However, the high values linked to the coefficient of determination are probably due to the statistical relationship between indices and vulnerability. The most significant limitations that have emerged from this experience are the 'black boxes' aspects of the component, and the absence of existing statistics that allow simple and effective comparison with other machine learning tools. Some potentialities emerged during this experimentation, such as the ease in implementing simple experiences and the usefulness of VPL as a learning medium for the main concepts linked to artificial intelligence towards the use of more robust frameworks (as Tensorflow and PyTorch). It is therefore possible to consider GH as a digital carnet where a domain expert in the field of drawing and surveying can sketch a prototype AI model to identify the problem. Once this is done, the model can be implemented working with AI experts or by the domain expert himself after a learning and training phase.

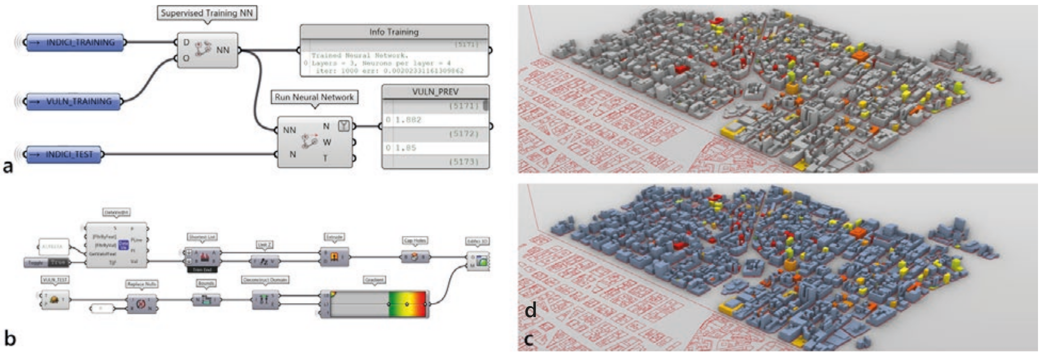


Fig. 2.
a) Training of the neural network with Dodo components;
b) VPL code to display prediction results;
Comparison of predicted (in blue) (c) vulnerability values and actual ones (in grey) (d).

Conclusions

There are still some open questions that constitute the next steps of the research. In particular, how to link the information produced in urban surveys to the classification of building types, how to predict the internal distribution pattern of building units using spatial syntax analysis (since it is not possible in many cases to access the interior spaces). Furthermore, from an economic point of view it is possible to envision a model that can support the prediction of the cost of seismic retrofit and/or demolition interventions.

Notes

[1] http://palermohub.opendatasicilia.it/ (15th February 2021).

[2] Dodo is a plug–in for Grasshopper developed by Lorenzo Greco. It is available on the online portal 'Food4Rhino' since 23/11/2015. Last version: 23/02/2019. Link: https://www.food4rhino.com/app/dodo (15th February 2021).

References

Boeykens Stefan, Neuckermans Herman (2013). Visual Programming in Architecture: Should Architects Be Trained as Programmers? In Tidafi Temy, Dorta Tomas (eds.). *Joining Languages Cultures and Visions: CAADFutures 2009*,

Caliò Ivo, Caponnetto Rossella, Ciatto Chiara, La Rosa Daniele (2018). A simplified methodology to assess seismic risk at district and building level. In Margani Giuseppe, Rodonò Gianluca, Sapienza Vincenzo (eds.). *Seismic and energy renovation for sustainable cities*. Gorizia: Edicom Edizioni.

Corradi Juri, De Fausti Fabrizio, Salvucci Gianluigi, Vitale Valerio (2014). Popolazione e vulnerabilità sismica. In *Giornate della ricerca in ISTAT* – 10-11 novembre 2014. Roma: Istat.

Haeberli Paul (1988). ConMan: a visual programming language for interactive graphics. In *SIGGRAPH Comput. Graph.*, 22 (4), pp. 103-111.

Halbert Daniel Conrad (1984). *Programming by Example*. PhD Thesis. Computer Science Division, Dept. of EE&CS, University of California, Berkeley.

Myers Brad Allan (1986). Visual programming, programming by example, and program visualization: a taxonomy. In *SIGCHI Bull.*, 17 (4), pp. 59-66.

Rutten David (2012). Programming, Conflicting Perspectives. I Eat Bugs for Breakfast. http://ieatbugsforbreakfast.wordpress.com/2012/04/01/programming-conflicting-perspectives/ (15th February 2021).

Tono Alberto (2018). BIMHOX: The Evolutionary In–formation Genes, AU Las Vegas 2018. https://www.autodesk.com/autodesk-university/class/BIMHOX-Evolutionary-Information-Genes-2018 (15th February 2021).

Vitrano Giovan Battista (2017). Mappa 3D – Vulnerabilità sismica degli edifici residenziali di Palermo. https://coseerobe.gbvitrano.it/mappa-3d-vulnerabilita-sismica-degli-edifici-residenziali-di-palermo.html (15th February 2021).

Zwierzycki Mateusz (2017). Why do Architects Code? TWF Conference 2017. https://www.youtube.com/watch?v=_4SBpq31L0M&t=1s&ab_channel=TheWayForwardCommunity (15th February 2021).

Author
Federico Mario La Russa, Dept. of Civil Engineering and Architecture, University of Catania, federico.larussa@phd.unict.it

 Isbn 9788835125280