

AUTOMATED ESTIMATION OF BUILDING HEIGHT THROUGH IMAGE PROCESSING

Pietro Carpanese¹, Marco Donà^{1,2}, and Francesca da Porto¹

¹ Depart. of Geosciences - University of Padova
Via G. Gradenigo, 6, 35131 Padova (IT)
pietro.carpanese@phd.unipd.it, francesca.daporto@unipd.it

² Earthquake Engineering Research & Test Center – Guangzhou University
Guang Yuan Zhong Rd. 248, 510405 Guangzhou, China
dona_marco@gzhu.edu.cn

Abstract

Seismic risk is one of the main problems in highly urbanized countries with a considerable seismic hazard like Italy. To assess seismic risk of the built heritage, it is necessary to deepen the knowledge of its components, vulnerability in particular. Vulnerability can be evaluated through mechanical methods, which require detailed information on parameters that may affect the seismic response. The implementation of such methods often results in time-consuming and expensive investigations, thus making the risk assessment analysis very cumbersome.

In order to make this process easier and faster, remote sensing algorithms can be taken into consideration. In this work, satellite images of areas of interest are automatically extracted via open source online maps, as well as some preliminary information about the buildings detected in the pictures. Afterwards, a filter is set in order to visualize only targeted building typologies (e.g., residential buildings), and street view images are obtained for each selected building. The images are then processed through feature extraction techniques, in order to predict the number of stories of the buildings.

The remote and automated retrieval of this feature, along with other meaningful parameters, could allow the association of a specific vulnerability level for each building, thus making on-site surveys unnecessary, with a remarkable reduction in time and costs.

Keywords: Remote Sensing, Satellite Images, Street View, Image Processing, Building Height, Seismic Vulnerability

1 INTRODUCTION

Seismic risk is one of the main problems in highly urbanized countries with a considerable seismic hazard like Italy. Even more, seismic risk mitigation represents a major challenge to reduce socio-economic impact of future earthquakes. To carry out seismic risk assessments, it is necessary to deepen the knowledge of the different components of this risk.

Seismic risk is defined as the combination of seismic hazard (probability that a particular earthquake occurs), exposure (number of people, quantity and value of buildings and goods) and seismic vulnerability. The latter can be described as the propensity of a building to suffer a certain level of damage after a seismic event. [1]

In order to assess seismic vulnerability of buildings, it can be useful to define fragility curves, which correlate ground motion with the damage suffered by the buildings. In literature, many approaches allow the calibration of fragility curves. Among others, analytical-mechanical approaches define fragility based on structural models in order to simulate the seismic behavior of buildings. [2-6]. However, when implementing mechanical methods to properly characterize the behavior of buildings at a territorial scale, it is always necessary to collect a big amount of structural data, such as material, construction period, height, as well as some more detailed geometric features. Especially when dealing with large districts or even entire towns, the investigation of these essential parameters often results in time-consuming and expensive surveys, making the risk assessment analysis very cumbersome [7-8].

In order to make this investigation process faster and easier, artificial intelligence techniques may be taken into consideration. For example, instance classification methods have already been used to classify buildings and some of their features using street view images [9-10]. Further information such as building age and category has also been obtained by analyzing morphology, neighborhood characteristics and urban patterns from topographic maps [11-16]. Some works have employed shadow detection techniques in satellite images, in order to estimate building heights [17-19], while others combine maps and street level images to perform the same task [20-23]. In general, it has become evident how artificial intelligence can be useful for elaborating models that extract building information, and consequently for carrying out risk management assessments [24].

In this paper, particular attention is paid to the retrieval of building footprints from satellite images of areas of different sizes, as well as some preliminary geometric and typological features. Subsequently, a methodology for estimating building heights (in terms of number of stories) from street view images is presented, where image processing and feature extraction techniques are used. The main steps of the algorithm are commented, along with the resolution of the main obstacles that can be encountered when analyzing street view pictures.

2 AUTOMATED RETRIEVAL OF BUILDING FOOTPRINTS

When seismic risk analyses are to be carried out at a territorial scale, an important element to assess is building exposure. This first step is necessary to understand how urbanized the area is (how many buildings there are), as well as to identify the building typologies. A good estimate of exposure, indeed, takes into account the different usages of buildings, as well as their construction material, construction period, etc.

For this aim, a Python code was developed, in order to retrieve building footprints given a specific satellite image.

To retrieve the desired satellite image, the script provides two input possibilities: the user can search a place by town/municipality, or can enter a pair of coordinates (latitude and longitude) and distance radius R . In the first case, the program will also return the centroid of the chosen town (in terms of latitude and longitude); in the second case, the program will extract

the name of the municipality, region and country to which the chosen area belongs. The code performs this task through the Static Maps service from the Mapbox API. Figure 1 shows two example of satellite images: Figure 1a shows an image extracted for an entire town (Padova), while Figure 1b displays an area centered in the point (45.409605, 11.888727) with a radius R of 300 meters.

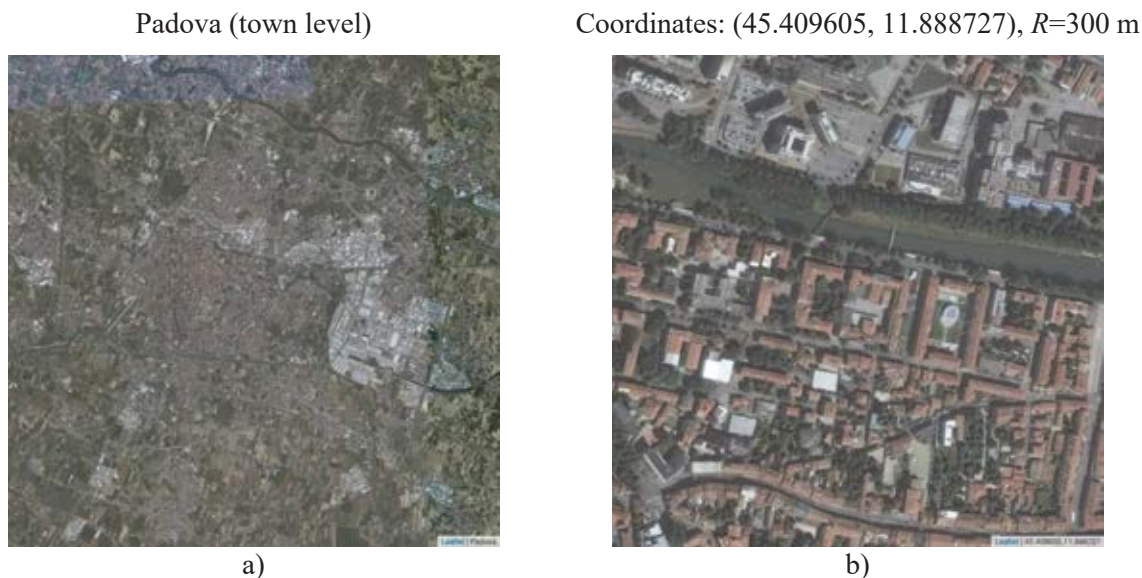


Figure 1: Satellite images for a) the town of Padova, b) an area centered in the point of coordinates (45.409605, 11.888727) with a radius of 300 meters

The second step of the code involves the acquisition of the building footprints that can be found in the satellite images. To do so, the OSM dependency is utilized, which provides access to OpenStreetMap services. Particularly, the key “building” is selected, in order to extract only this type of objects from the map. The two maps in Figure 2 show the building footprints related to the satellite images shown in Figure 1.

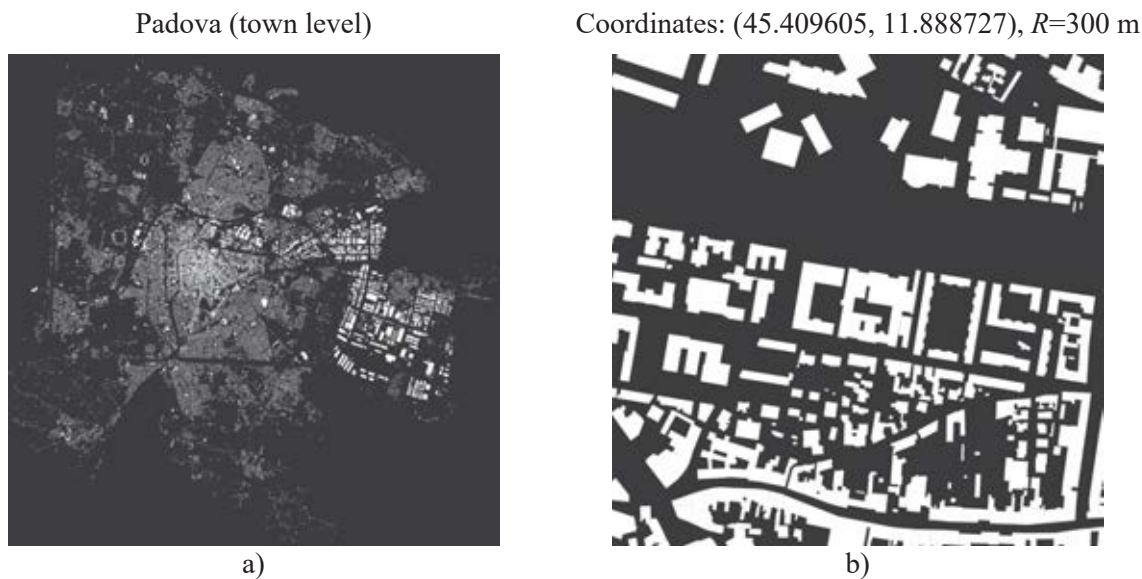


Figure 2: Building footprints for a) the town of Padova, b) an area centered in the point of coordinates (45.409605, 11.888727) with a radius of 300 meters

This section of the code not only retrieves the building footprints, but also retains their floor area, their centroid (latitude-longitude) and the labels of each building (such as “accommodation”, “commercial”, “religious”, etc). All the data obtained are stored in a DataFrame, where each row represents a building; they can be accessed and edited via GeoPandas, an open source project that helps manage geospatial data in Python. Figure 3 shows the building footprints from Figure 2b, with different colors according to their label.



Figure 3: Building footprints and their labels

In addition, the way in which data are stored makes it easy to filter them. As an example, only residential buildings and their outbuildings are shown in colors in Figure 4, while other building typologies are plotted in gray.



Figure 4: Building footprints and their labels, where only residential buildings and outbuildings are kept

As said before, for each building the centroid is known. Thanks to the Google Street View API, it is possible to automatically obtain the façade picture of a desired building, given its centroid coordinates. The user can select the size of the output picture, the pitch (vertical angle) and the heading (horizontal angle). In this work, the default size was chosen, i.e., 640x640 pixels. Also, no heading was specified: each time, a value is calculated that directs the camera towards the specified location, from the point at which the closest photograph was taken. For what concerns the pitch, an angle of 5° was selected, so that even taller buildings can fit in the

image taken (the default value would be 0° , which means flat horizontal). In Figure 5, some examples of street view images associated to the area under consideration are shown. The code could easily acquire the street view pictures for all the building footprints in the image, in a fairly short computation time.



Figure 5: Examples of street view images obtained from the coordinates of the building footprints

3 AUTOMATED ESTIMATION OF BUILDING HEIGHTS

The workflow described in §2 leads to a good estimate of building exposure in a specific area, in terms of number of buildings, building typology and total built-up area.

As said before, however, such information is not sufficient to predict the seismic vulnerability and risk level of a urban district, since more building parameters should be evaluated. In particular, previous works have shown that a good estimate of vulnerability can be obtained by knowing at least the number of stories, the material and the construction period of the buildings examined [5].

In this paper, the variable “number of stories” is examined. Specifically, an algorithm that automatically retrieves the number of floors of buildings from street view images was developed. Image processing and feature extraction techniques were implemented in this algorithm.

3.1 Building detection with bounding boxes

The first step in this algorithm is the correct identification of buildings in the street view image. To perform this task, other works have used Convolutional Neural Networks (CNNs) that are able to detect and classify façade structures from street view pictures [9].

In this specific case, however, a complete detection of buildings is not necessary. In fact, to estimate the number of stories, only a single vertical strip of windows could be sufficient. For this reason, a script that selects the appropriate bounding box was developed. A bounding box is a rectangle that fully encloses an object in a digital image; in this case, the bounding box is set to include one or two vertical strips of windows (optionally including the door), going from the bottom of the building (e.g pavement line) to the rooftop (e.g skyline). In this way, the rest of the code will only have to deal with a smaller portion of the detected building, thus reducing the computation time and possible overlapping errors.

The method used to detect a bounding box relies on the Hough transform. The Hough transform is a feature extraction technique often used in digital image processing, whose main purpose is the identification of lines in the image [25].

Firstly, only the vertical Hough lines are taken into consideration, with a range of slopes between -15° and $+15^\circ$, where a 0° represents a perfectly vertical line (green lines in Figure 6b). By doing so, vertical strips of windows can be easily defined.

Secondly, the ground and the sky must be detected. Also in this step, the Hough transform is utilized, keeping only the lines whose slope is between 60° and 90° (both positive or negative). These values were chosen to make it possible to detect the base and the rooftop of a building, even when the street view image is not perfectly frontal. For detecting the skyline (blue line in Figure 6b), only the top half of the street view image is considered, while for the bottom line of the building (red line in Figure 6b) only the bottom half is kept. The fact that the street view images are acquired with pitch angles of 5° helps to remove part of the pavement, which is an irrelevant element for creating the bounding box.

The code was written in MATLAB, and more information about the parametric representation of lines with the Hough transform can be found in the algorithms section of the mathworks page.

In Figure 6c, it can be seen that from a rather complex street view image (with more than one building, a car, a gate, trees), the algorithm is able to crop only the part of the picture that will be useful for the estimation of the number of stories.

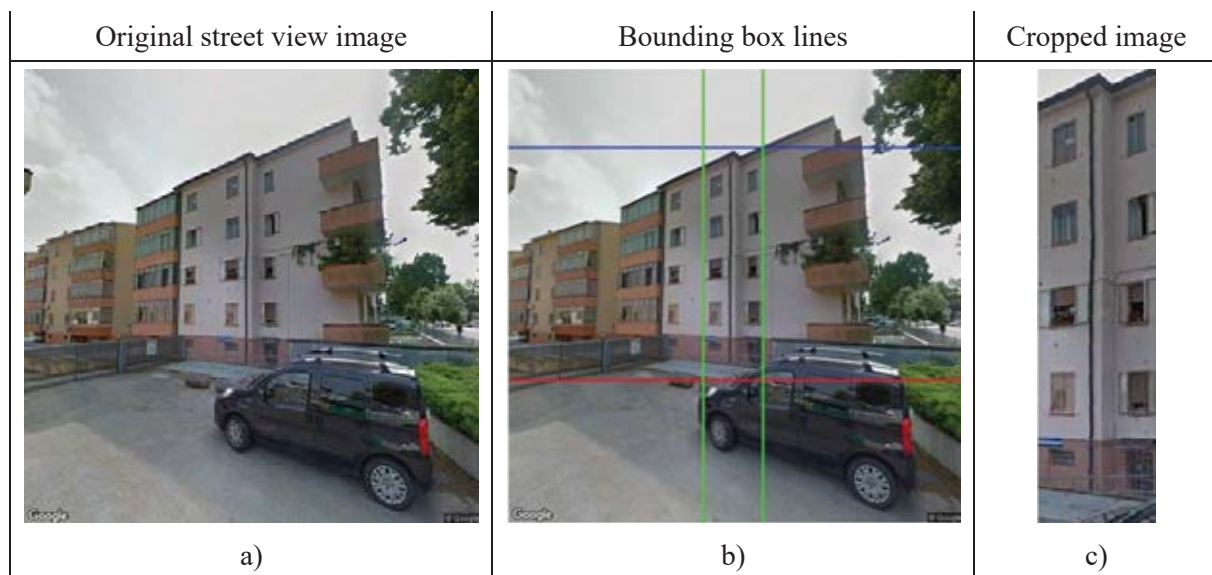


Figure 6: Example of a bounding box

3.2 Feature extraction for estimating the number of stories

Now that only a meaningful part of the street view image has been selected, the algorithm can proceed with the estimation of the number of stories. The key concept of this step is that the top and bottom of every door and window (or row of windows) must be detected, so that they can be associated with the presence of a floor.

To begin with, the raw Google Street View image has to be processed. Firstly, the original picture is turned into a grayscale image, and then an edge detection method is implemented. Edge detection is an image processing technique for finding the boundaries of objects within images, which works by detecting discontinuities in brightness. This technique is commonly used for image segmentation and data extraction. In this work, a Canny edge detection method

is implemented [26]. This method finds edges by looking for local maxima of the gradient of the grayscale image, which is calculated using the derivative of a Gaussian filter. The Canny method is one of the most suitable to detect true edges. The result is a binary (black and white) image, as shown in the example of Figure 7c for a bounding box image.

Secondly, morphological operations are applied to the binary image. These operations can process images based on shapes, so that each pixel in the image is adjusted based on the value of other pixels in its neighborhood. In particular, the following morphological operations are used: *skeleton*, which removes pixels on the boundaries of objects but does not allow objects to break apart; *thin*, which thins objects to lines; *spur*, which removes pixels starting from the end of any branch as well as the two ends of the main skeleton; and lastly *clean*, which removes isolated pixels. Essentially, all these operations allow a simplification of the binary image, where only the main lines are kept (as can be seen in Figure 7d).

The final step is the window (and consequently floor) detection. Once again, the Hough transform was used to detect horizontal lines, as described in §3.1. Out of all the lines that are found, the ones that are too close to each other (i.e., at a distance of less than 20 pixels) are grouped together, and only one of them is kept. This solution was adopted in order to avoid detecting the same line twice, since the border of a window may be associated with multiple Hough lines. In addition, the script eliminates intersecting lines, which are usually caused by obstacles in front of the actual window or door, such as the branches of a tree.

Figure 7 shows the general workflow that has just been explained. In Figure 7a, the original street view image is shown, as obtained from Google Street View. Figure 7b shows the bounding box that selects only the main door and one strip of windows. Figure 7c shows the result of applying the Canny Edge method to the bounding box image, while in Figure 7d morphological operations are applied to the binary image. Lastly, Figure 7e represents the feature extraction step, where the Hough transform technique is used to identify the lines delimiting doors and windows.

Looking at Figure 7e, the estimation of number of stories is immediate, since every couple of lines represents a different floor. Moreover, the code automatically computes this parameter, by taking into account the final number of lines, dividing it by two and rounding this value down.

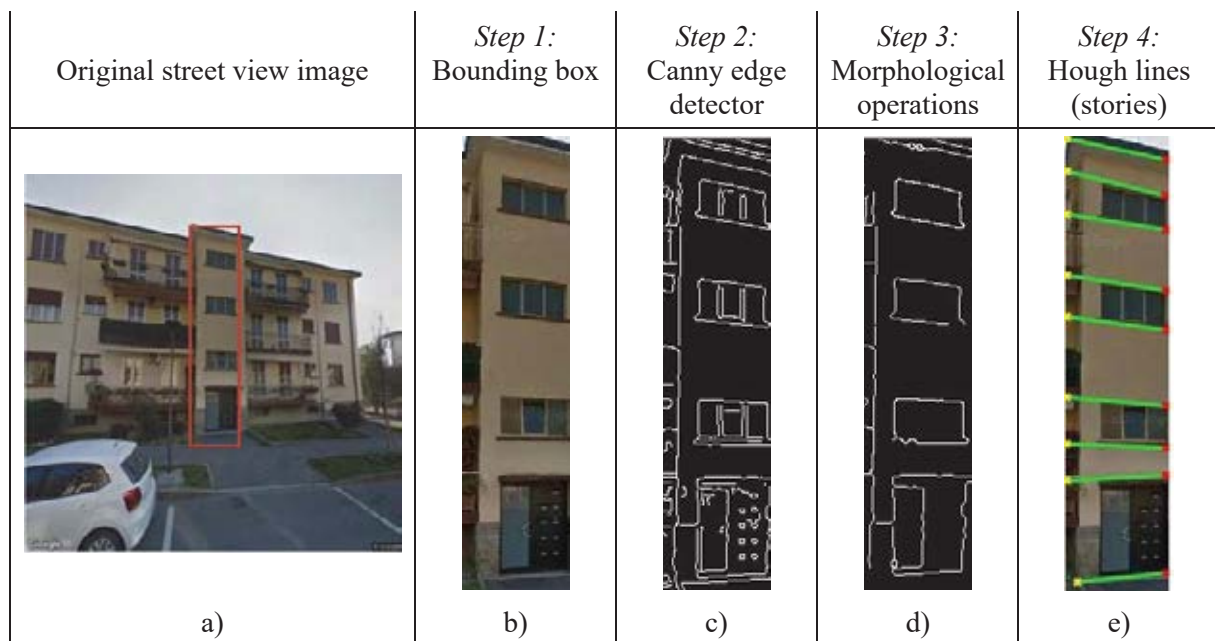


Figure 7: Main steps of the algorithm to estimate the number of stories of a building from street view images

3.3 Particular case: bounding box for hidden buildings

The code described in §3.2 works with a good degree of accuracy when the building is completely visible, and the façade can be seen from a frontal or near-frontal perspective. However, many Google Street View images contain pictures of buildings that are partially hidden or covered by obstacles. In order to cope with this problem, some adjustments were made in the algorithm. Here, the case where a tree blocks part of the picture is commented.

In the example provided in Figure 8a, tree branches and leaves occupy almost the entire top half of the picture. When applying the Canny edge detector to this type of images, a lot of noise (in terms of blobs and irregular shapes) is generated in association with these objects, as can be seen in Figure 8b. This may affect the performance of the code, since the Hough transform method may mistake lines belonging to the branches for window lines.

Fortunately, the building that has to be detected (the one in the middle) is not concealed, and the base and rooftop are both visible behind the tree, as well as the door and the windows. For this reason, some specific morphological operations can be performed, so that the lines and dots belonging to the leaves are deleted, and the algorithm is able to operate only with lines associated to the actual building. In particular, the operation called *branchpoints* is used, which finds lines connected to the main objects of the image. Subsequently, the branchpoints are enlarged, in order to really separate the lines. Then, the branchpoints are erased from the original binary image to leave just the single, unbranched lines, as shown in Figure 8c. Lastly, the blobs that are formed by less than n pixels can be deleted (in this work, a value of 30 pixels was chosen), and the result is shown in Figure 8d. In this last image, it is clear how the noise due to the tree can be almost completely removed, maintaining only the main lines of the building under consideration. Figure 8e shows the bounding box related to the building, and it can be noticed how the algorithm can detect the façade without being influenced by the tree or other elements.

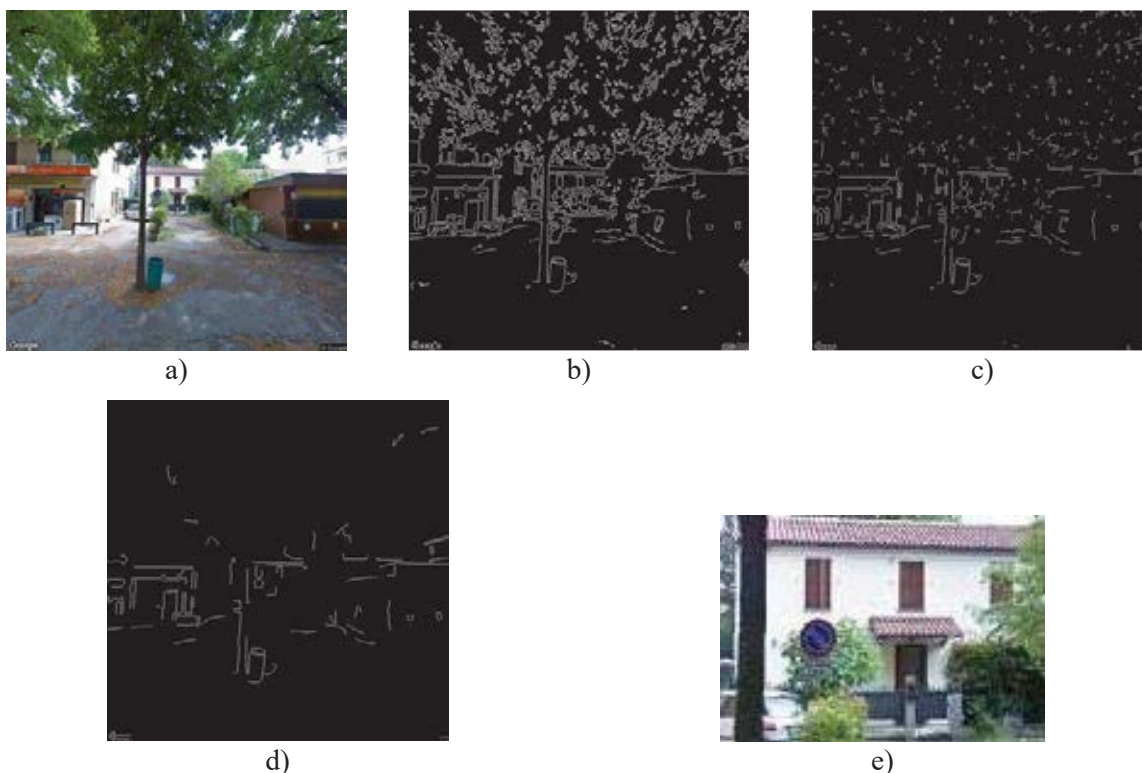


Figure 8: Example of bounding box for a street view image with an obstacle (tree)

4 CONCLUSIONS

- An algorithm that automatically retrieves satellite images from a pair of coordinates or a town name was discussed. The code also returns the building footprints that can be found in the image, together with their coordinates, areas, and labels. These data can be edited, and the resulting map can be filtered. The following part of the algorithm uses the coordinates of the buildings found in the area to acquire street view images.
- A second algorithm was presented, to estimate the number of stories of the buildings found from the previous code. The street view images obtained before are the input of this algorithm. Firstly, the code creates a bounding box that selects only the part of the building that is sufficient to estimate its number of stories. Secondly, the number of stories is evaluated through image processing and feature extraction techniques, particularly the Canny edge detection method, morphological operations and the Hough transform. In addition, some particular cases where the building is hidden by obstacles were treated.
- The work presented in this paper is in phase of validation. The code works best for buildings that can be seen from a frontal perspective, although some expedients were adopted for images that do not have the same characteristics. However, a more systematic validation should be carried out, involving different areas, different territorial scales, and different building typologies.
- This work fits into the wider context of automated seismic risk assessment. In order to carry out risk analysis, the estimation of the height parameter is not sufficient to achieve an acceptable level of accuracy. Other variables will have to be evaluated, first of all construction period and material. Once all these parameters are known, it will be possible to estimate seismic vulnerability more accurately and automatically at a territorial scale. Therefore, this algorithm could represent a useful tool for authorities in the field of seismic risk management, and it could make on-site surveys unnecessary, with a remarkable reduction in time and costs.

ACKNOWLEDGMENTS

This work is part of the European project BESTOFRAC. The authors would like to offer special thanks to the Institut für Strukturmechanik (ISM) of the Bauhaus-Universität in Weimar (Germany).

REFERENCES

- [1] M. Dolce, A. Prota, B. Borzi, F. da Porto, S. Lagomarsino, G. Magenes, C. Moroni, A. Penna, M. Polese, E. Speranza, G.M. Verderame, G.Zuccaro, Seismic risk assessment of residential buildings in Italy. *Bulletin of Earthquake Engineering*, 2020.
- [2] M. Donà, P. Carpanese, V. Follador, F. da Porto, Derivation of mechanical fragility curves for macro-typologies of Italian masonry buildings. M. Papadrakakis, M. Fragiadakis eds. *7th ECCOMAS Thematic Conference on Computational Methods in Structural Dynamics and Earthquake Engineering (COMPDYN 2019)*, Crete, Greece, June 24-26, 2019.

- [3] S. Lagomarsino, S. Cattari, Fragility functions of masonry buildings, (Chapter 5), K. Pitilakis, H. Crowley, A.M. Kaynia eds. *SYNER-G: Typology Definition and Fragility Functions for Physical Elements at Seismic Risk, Vol. 27*, 111- 156, Springer 2014.
- [4] B. Borzi, M. Faravelli, A. Di Meo, Application of the SP-BELA methodology to RC residential buildings in Italy to produce seismic risk maps for the national risk assessment. *Bulletin of Earthquake Engineering*, 2020.
- [5] M. Donà, P. Carpanese, V. Follador, L. Sbrogiò, F. da Porto, Mechanics-based fragility curves for Italian residential URM buildings. *Bulletin of Earthquake Engineering*, 2020.
- [6] F. da Porto, M. Donà, A. Rosti, M. Rota, S. Lagomarsino, S. Cattari, B. Borzi, M. Onida, D. De Gregorio, F.L. Perelli, C. Del Gaudio, P. Ricci, E. Speranza. Comparative analysis of the fragility curves for Italian residential masonry and RC buildings. *Bulletin of Earthquake Engineering, Special Issue: Seismic Risk Assessment in Italy*, 2021.
- [7] M. Vettore, M. Donà, P. Carpanese, V. Follador, F. da Porto, M.R. Valluzzi, A Multilevel Procedure at Urban Scale to Assess the Vulnerability and the Exposure of Residential Masonry Buildings: The Case Study of Pordenone, Northeast Italy. *Heritage*, **3**(4), 1433–1468, 2020.
- [8] G.P. Campostrini, S. Taffarel, G. Bettiol, M.R. Valluzzi, F. da Porto, C. Modena, A Bayesian approach to rapid seismic vulnerability assessment at urban scale, *International Journal of Architectural Heritage*, **12**, 36-46, 2017.
- [9] J. Kang, M. Körner, Y. Wang, H. Taubenböck, X.X. Zhu, Building instance classification using street view images. *ISPRS Journal of Photogrammetry and Remote Sensing*, **145**, 44–59, 2018.
- [10] J. Castagno, E. Atkin, Roof shape classification from LiDAR and satellite image data fusion using supervised learning. *Sensors (Switzerland)*, **18**(11), 2018.
- [11] J.F. Rosser, D.S. Boyd, G. Long, S. Zakhary, Y. Mao, D. Robinson, Predicting residential building age from map data. *Computers, Environment and Urban Systems*, **73**, 56–67, 2019.
- [12] R. Hecht, G. Meinel, M. Buchroithner, Automatic identification of building types based on topographic databases – a comparison of different data sources. *International Journal of Cartography*, **1**(1), 18–31, 2015.
- [13] M. Fleischmann, A. Feliciotti, O. Romice, S. Porta, Morphological tessellation as a way of partitioning space: Improving consistency in urban morphology at the plot scale. *Computers, Environment and Urban Systems*, **80**, 2020.
- [14] G. Boeing, Spatial information and the legibility of urban form: Big data in urban morphology. *International Journal of Information Management*, **56**, 2021.
- [15] M. Wieland, M. Pittore, Performance evaluation of machine learning algorithms for urban pattern recognition from multi-spectral satellite images. *Remote Sensing*, **6**(4), 2912–2939, 2014.
- [16] S. Steiniger, T. Lange, D. Burghardt, R. Weibel, An approach for the classification of urban building structures based on discriminant analysis techniques. *Transactions in GIS*, **12**(1), 31–59, 2008.
- [17] G. Liasis, S. Stavrou, Satellite images analysis for shadow detection and building height estimation. *ISPRS Journal of Photogrammetry and Remote Sensing*, **119**, 437–450, 2016.

- [18] A. Comber, M. Umezaki, R. Zhou, Y. Ding, Y. Li, H. Fu, H. Jiang, A. Tewkesbury, Using shadows in high-resolution imagery to determine building height. *Remote Sensing Letters*, **3**(7), 551–556, 2012.
- [19] P.L.N. Raju, H. Chaudhary, A.K. Jha, Shadow analysis technique for extraction of building height using high resolution satellite single image and accuracy assessment. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, **40**(8), 1185–1192. 2014.
- [20] J. Yuan, A.M. Cheriyyadat, Combining maps and street level images for building height and facade estimation. *Proceedings of the 2nd ACM SIGSPATIAL Workshop on Smart Cities and Urban Analytics, UrbanGIS 2016*, Burlingame, California, United States, October 2016.
- [21] E.D. Diaz Plata, H Arguello, An algorithm to estimate building heights from Google street-view imagery using single view metrology across a representational state transfer system. *Proceedings of the SPIE, Volume 9868*, Baltimore, Maryland, United States, April 2016.
- [22] G.C. Iannelli, F. Dell’Acqua, Extensive exposure mapping in urban areas through deep analysis of street-level pictures for floor count determination. *Urban Science*, **1**(2):16, 2017.
- [23] Y. Zhano, J. Qi, R. Zhang, CBHE: Corner-based building height estimation for complex street scene images. *WWW '19: The World Wide Web Conference*, 2436-2447, San Francisco, California, United States, May 2019.
- [24] C. Wang, Q. Yu, K.H. Law, F. McKenna, S.X. Yu, E. Taciroglu, A. Zsarnóczyay, W. Elhaddad, B. Cetiner, Machine learning-based regional scale intelligent modeling of building information for natural hazard risk management. *Automation in Construction*, **122**, 2021.
- [25] R.O. Duda, P.E. Hart, Use of the Hough transformation to detect lines and curves in pictures. *Communication of the ACM*, **15**(1), 1972.
- [26] J.F. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **8**(6), 679-698, 1986.