COMPUTATIONAL TOPOLOGY TO MONITOR HUMAN OCCUPANCY

Paolo Barsocchi, Pietro Cassará , Daniela Giorgi, Davide Moroni, Maria Antonietta Pascali

Institute of Information Science and Technologies - CNR, Via G. Moruzzi 1, 56124 Pisa, ITALY

ABSTRACT

The recent advances in sensing technologies, embedded systems, and wireless communication technologies, make it possible to develop smart systems to monitor human activities continuously. The occupancy of specific areas or rooms in a smart building is an important piece of information, to infer the behavior of people, or to trigger an advanced surveillance module. We propose a method based on computational topology to infer the occupancy of a room monitored for a week by a system of low-cost sensors.

Index Terms— WSN, surveillance, occupancy, human behavior, computational topology, persistent homology

1. INTRODUCTION

Improving the energy efficiency of buildings is a lively research topic. Technologies for the monitoring and consequent intervention on the environment not only help saving resources, but also improve the well-being perceived by building occupants [1]. One of the simplest policies to improve energy-efficiency while respecting the well-being of occupants, is the automatic regulation of the usage of energy based on the occupancy of the building spaces [2]. If no users are detected in the monitored space, inactive devices can be disconnected; environmental setpoints for air conditioners, heaters, and humidifiers can be decreased; lights can be lowered or turned off.

The key requisite for implementing occupancy-driven applications is an accurate, inexpensive, and non-intrusive method for monitoring occupancy. Non-intrusive means that the user is not required to have smartphones or RFID identification systems, which provide information about its presence within the space. Usually, such systems are referred to as *device-free*, and exploit the networks of acoustic, inertial, environmental and power sensors already installed in the buildings [3].

Recently, many authors proposed to analyse data from multiple sensors (electric power meters, accelerometers and noise meters) to identify the user's presence or absence in a space [4, 5]: variations of sensor time series, due to the arrival or exit of a user from a space, are analyzed to infer a classification template for the user's presence/absence. Conversely, [6, 7, 8] address the problem of identifying user activities by analyzing the spectral fingerprint of the energy consumption due to the used appliances during their activities. [9] investigate about the possibility of identifying the user activity by analyzing energy consumption patterns from electrical appliances.

Device-free sensing of building occupancy poses many challenges, as any sensor shows a noisy behavior under certain conditions. For example, inertial sensors can cause spurious detection deriving from pets movements; though CO2 sensors are less sensitive to their placement and to external events, their efficiency is directly related to the ventilation of the building. Therefore, robustness to noise is a key issue for device-free sensor system. In this work, we present an algorithm for extracting topological features from sensor time series, to support the implementation of a device-free detection algorithm of presence/absence. Basing on the theory of Persistent Homology, we evaluate the relevance of topological features as their *persistence* (a topological event's lifespan) in the so-called *barcodes* [10]. The assumption is that persistent features identify important events, while features with a shorter lifespan correspond to noise. The result is a topological fingerprint of the user's presence, which can be used in an unsupervised approach for state classification. The main advantages brought by Persistent Homology are low computational cost, and high resilience to noisy data.

2. TOPOLOGICAL DATA ANALYSIS

Topology is a branch of mathematics dealing with *qualitative* information of an object: topology looks at the intrinsic and global properties of an object, such as its shape [11]. Recently, Topological Data Analysis (TDA) methods are increasingly used to investigate and characterize multidimensional datasets. The research on TDA was recently boosted by the introduction of the Persistent Homology (PH) theory [12], along with fast algorithms for its computation [13] and efficient implementations [14]. The application scenarios include shape and texture analysis [15], biological and molecular data analysis [16], sensor networks [17], image and signal processing [18, 19, 20]. In what follows, we offer a brief intuition on

For the time series used to validate the results presented in this paper, we give thanks to the framework of the Renewable Energy and ICT for Sustainability Energy project, supported by the Italian Ministry of University and Research (MIUR) through the project Active Ageing at Home (CTN01 00128 297061)

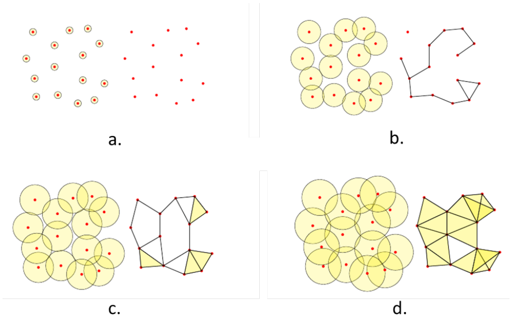


Fig. 1. An example of the Rips-Vietoris complex associated to a 2D point could, with the filtration parameter d induced by the Euclidean distance betweenpints. At $d = 0$ the complex is made of the 0-points, as shown in a.; increasing d, edges, faces and tetrahedra enter the simplical complex.

PH; we refer the reader to [21] for a rigorous treatment of the subject.

The core idea of Persistent Homology is to represent data as *filtered simplicial complexes*. Given a simplicial complex (for example, a triangle mesh), filtering the complex means defining a rule to build the complex as a sequence of nested sub-complexes. An example of filtered complex is the the Vietoris-Rips complex, whose construction is shown in the example of Figure 1: at the first stage of the filtration, only the nodes are included; then, in subsequent steps, the edges in the complex appear ordered by the distance between their endpoints (the filtration parameter). There are many viable choices on how to filter a complex associated to a dataset: this is one of the main strengths of PH-based approaches. Once the dataset is encoded as a filtered simplicial complex, one evaluates the birth and death of topological events (*homology classes*) while growing the filtration parameter: for example, when connected components appear and when are merged, when holes are created and when are closed off, and so on. The lifespan of these events is stored in a stable invariant: the $i-$barcode. A barcode in a given dimension i is a collection of horizontal bars in a plane: the horizontal axis corresponds to the parameter to filter the complex, while the vertical axis represents an arbitrary ordering of homology generators in dim i (see Figure 2 for an example of barcodes in dim 0 and 1). The length of each bar is interpreted as the lifespan or *persistence* of the corresponding generator: short bars are interpreted as noise, while long bars as important topological features.

In this paper, we compute the persistence barcodes on time series from multiple sensors placed in a room, to detect the points in time when the room is occupied. The assumption is that room occupancy is reflected by some topological features in PH. To compute the barcodes, the first step is to discretise the sensor time series and represent them as filtered complexes, as described in the next Section.

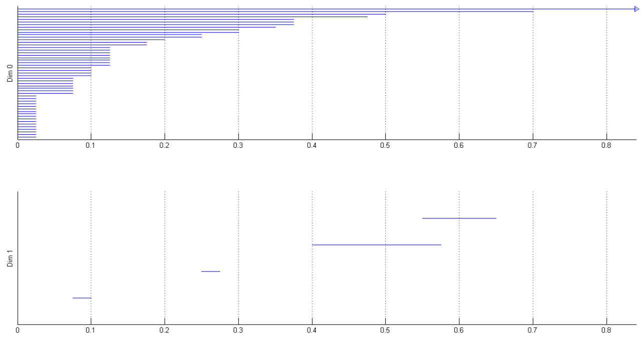


Fig. 2. An example of two barcodes (dim 0 and 1) from the experiment described in Section 3: they are computed from F_{59}, with $W = 68$

$$\begin{bmatrix} \mathbf{S_{1,1}} & \cdots & \mathbf{S_{1,K}} \\ \vdots & \vdots & \vdots \\ \mathbf{S_{W+1,1}} & \vdots & \mathbf{S_{W+1,K}} \\ \vdots & \ddots & \vdots \\ \mathbf{S_{j+W,1}} & \vdots & \mathbf{S_{j+W,K}} \\ \vdots & \vdots & \vdots \\ \mathbf{S_{M,1}} & \cdots & \mathbf{S_{M,K}} \end{bmatrix} \mathbf{W}$$

Fig. 3. Matrix Representation of the Data

3. FEATURE EXTRACTION ALGORITHM

Given K different sensor time series, each of length M, we represent them with a matrix (Figure 3): each column corresponds to the time series of a single sensor, whereas each row corresponds to all sensor values sampled at a given time. Then, we cluster rows into sliding windows F_i [19]: each F_i corresponds to W time samples of all K sensors, with each window sharing $W/2$ samples with the previous one. Therefore, each F_i can be seen as a point cloud, made of W points in the Euclidean space $\mathbb{R}^K$, where the coordinates correspond to the output of the different sensors.

For each point cloud F_i associated with a sliding window, we compute the Vietoris-Rips complex filtered by the Euclidean distance between points, and compute the corresponding barcodes. The analysis of the barcodes then gives insights on the likelyhood of a window to correspond to a period when the room is occupied. In particular, we analyse the barcodes through a set of descriptors derived by the topological features summerized in each barcode [20]: the number of $0-$ and $1-$ topological events; the number of $0-$ and $1-$ topological *long* events (i.e., events with lifespan greater than a fixed threshold: τ_0 in dimension 0 and τ_1 in dimension 1); and the average lifespan in dimension 0 and 1.

The feature extraction algorithm is summarised in Algorithm 1. The computations have been carried out using JavaPlex [22].

The set of descriptors is evaluated as a predictor of the

Algorithm 1 PH-based Algorithm for Feature Extraction

```
Step 1.
    a. Group data in blocks of length $W$, overlapping of $W/2$ each other
    b. Compute the associated set of point clouds $F_i$
    c. Mean-center and normalize each $F_i$

Step 2.
    For each $F_i$:
    a. Compute the associated barcodes in dimension 0 and 1 (i.e. $B_{0,i}$ and $B_{1,i}$)
    b. Count the topological events in $B_{0,i}$ and $B_{1,i}$
    c. Count the long topological events
    d. Compute the average lifespan for each barcode
```

occupancy of the room. The detection procedure works as follows: *(i)* the window F_i for which the descriptor reaches its highest values is selected; *(ii)* this selection is performed over all the descriptors; *(iii)* the intersection among the set of selected windows is evaluated. The detection of presence occurs in the windows belonging to the intersection; and the absence in the remaining windows.

4. EXPERIMENTS

The following subsections describe our scenario and the results we got in our preliminary study.

4.1. Experimental set-up

The detection of the user's presence was performed using time series acquired by three sensors: motion, acoustic, and a power meters sensor. Acoustic and motion sensors are characterized by a binary output: they return 1 if motion, or respectively high level of environmental noise, is detected, and 0 otherwise. The output of the power meter is a scalar that ranges from 0 to the level of consumed electric power.

The sensors' time series were acquired at an average sampling rate of 1 per minute, through a monitoring system installed in an office at the National Research Council of Italy (CNR) in Pisa. The office is 25 m^2 and it is used by only one employee. The typical working day at CNR ranges from 9.00 AM to 6.00 PM (Monday - Friday). Lamps, PCs and other appliances such as printers, kettles or coffee machines, may be present in the room.

The sensors deployed in the office communicate via ZigBee protocol with a gateway based on the Raspberry PI board which runs an integration middleware for sensing information. The data collected by the gateway are stored on a cloud infrastructure hosted at CNR. The cloud is organized as a set of virtual machines based on an VMware ESX Server providing three kinds of services: storage, visualization through an interactive dashboard, and tools for the analysis of data.

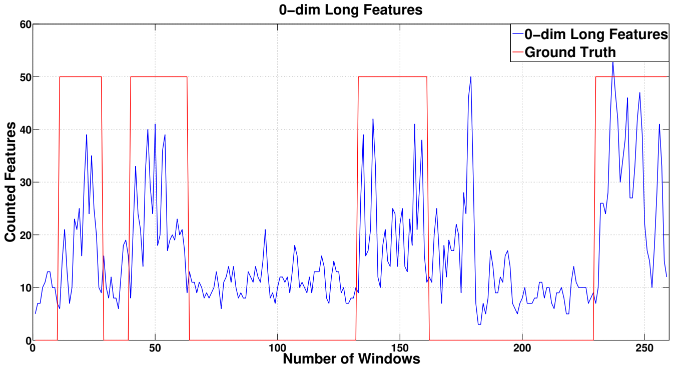


Fig. 4. 0-dim Long Features. The graph of the number of features (blue) is shown together with the ground truth graph (red)

The data have been collected for one week asking the occupant to register the ground truth of the room occupancy.

The three time series were temporally aligned: a week of acquisition produced 8840 samples for each time series. So, the data matrix in Figure 3 has $M = 8840$ rows and $K = 3$ columns. Rows have been grouped in sliding windows of size: $W = 18, 34, 68$. For each value of the window size W, we obtain a sequence of point clouds F_i in $\mathbb{R}^3$, with the corresponding Rips-Vietoris complex and persistence barcodes.

The number of long topological features, for each F_i, was counted using different values for both thresholds τ_0 and τ_1. Precisely, we have characterized topological events on a lattice of points (τ_0, τ_1), both ranging in the interval $[0.1, 0.9]$ with step size 0.2.

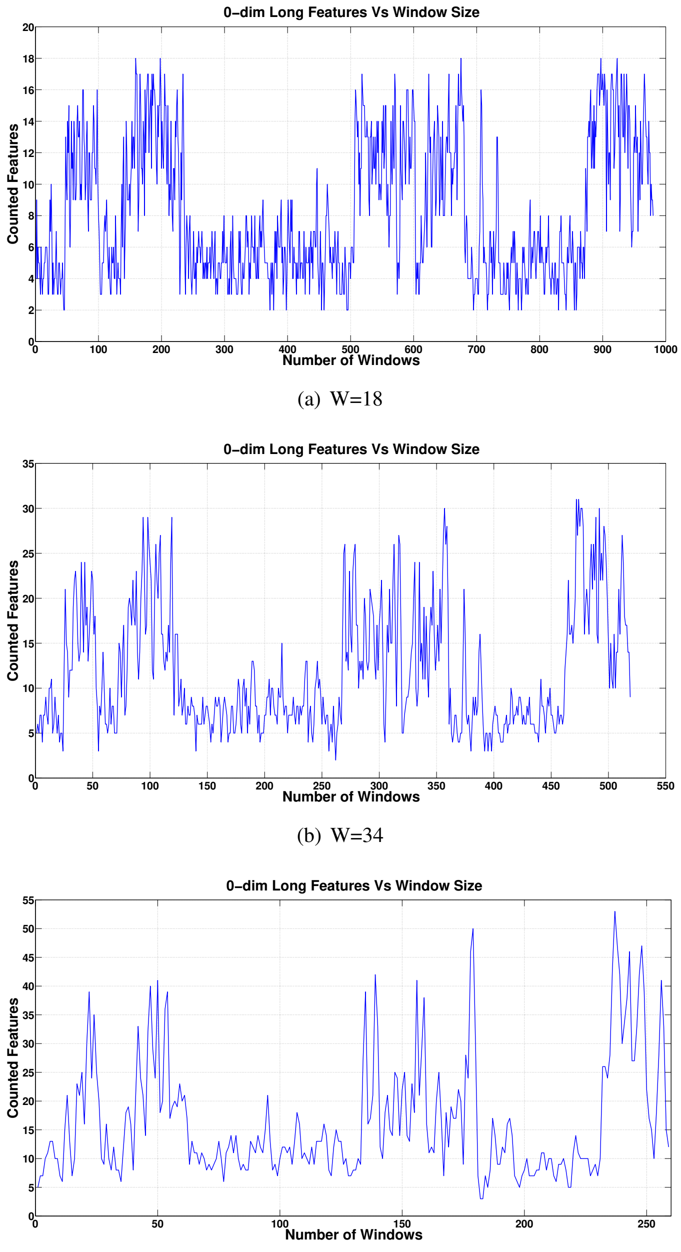
4.2. Results

Relation between descriptors and room occupancy

Figure 4 shows the number of counted features for each F_i, with $W = 68$. The comparison of this graph with the ground truth suggests that the greater number of features is obtained for those windows corresponding to the presence of the user in the office. In general, features in dim 0 showed better correlation with room occupancy than features in dimension 1. This is probably due to the fact that 0-dimensional features are linked to the correlation between samples within the windows considered, while features in dim 1 are indicators of possible periodic pattern in the signal time series, which are negligible in our data.

Dependence on the window size

Figure 5 shows the dependence of the $0-$*dim Long Features* on the window size W. It shows that the more samples are included within the same window, the more information can be extracted: increasing the window size, the number of features counted in windows corresponding to the user's presence increases. On the other hand, a smaller window size would imply a lower number of features and may cause the loss of useful information.



(a) W=18

(b) W=34

(c) W=68

Fig. 5. 0-dim Long Features vs. Window Size: each graph shows the number of features counted in the corresponding sliding window of size 18 in (a), 34 in (b), and 68 in (c). Note that increasing the window size results in a larger number of features.

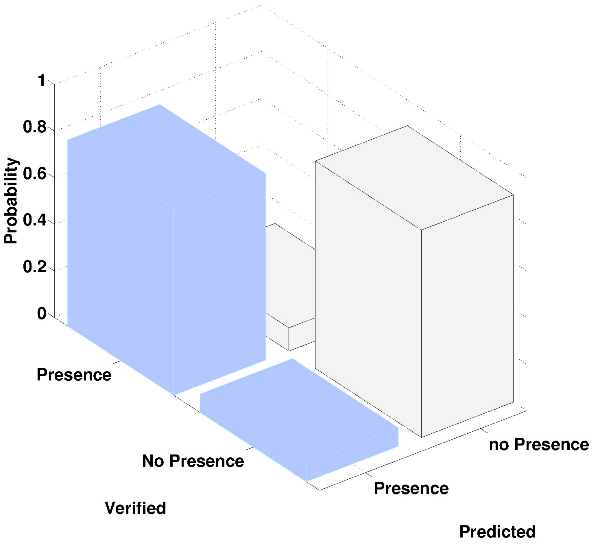


Fig. 6. Performance of the prediction algorithm: confusion matrix after one week

Prediction performance

Figure 6 shows the prediction performance obtained with the descriptor extracted from sensor time series with the values $W = 68$ and $\tau_0 = 0.1$, which provided the best set of descriptors for the detection procedure. The values 'Presence' and 'no Presence' in the axes 'Verified' are the ground truth, as registered by the room occupant; values 'Presence' and 'no Presence' in the axes 'Predicted' are those obtained by our detection algorithm. Figure 6 shows that the detection algorithm detects a 'Presence', which is confirmed by the ground truth. Instead, with a probability of 88% the detection algorithm detects a 'no Presence', which is again confirmed by the ground truth. The algorithm performance in our preliminary study is promising compared to the results in [3], considering that their period of experimentation is quite longer (one month vs one week).

5. CONCLUSIONS AND FUTURE WORK

The results reported in the previous section show that PH-based methods are useful to encode the information from low-cost monitoring systems about human activity. A further analysis may be carried out in order to understand if the proposed set of topological features may characterize more precisely human activity, not limiting to the occupancy of a room. Viable options for developing a topology-based activity detection may be: increasing the number of low-cost sensors used for the data acquisition, increasing the computation of PH to dimensions > 1, and complementing topological data analysis with adaptive detection techniques.

6. REFERENCES

[1] Paolo Barsocchi, Erina Ferro, Luigi Fortunati, Fabio Mavilia, and Filippo Palumbo, "Ems@ cnr: An energy monitoring sensor network infrastructure for in-building location-based services," in *High Performance Computing & Simulation (HPCS), 2014 International Conference on*. IEEE, 2014, pp. 857–862.

[2] Paolo Barsocchi, Mario GCA Cimino, Erina Ferro, Alessandro Lazzeri, Filippo Palumbo, and Gigliola Vaglini, "Monitoring elderly behavior via indoor position-based stigmergy," *Pervasive and Mobile Computing*, vol. 23, pp. 26–42, 2015.

[3] P. Barsocchi, A. Crivello, M. Girolami, F. Mavilia, and E. Ferro, "Are you in or out? monitoring the human behavior through an occupancy strategy," in *2016 IEEE Symposium on Computers and Communication (ISCC)*, June 2016, pp. 159–162.

[4] W. Kleiminger, C. Beckel, T. Staake, and S. Santini, "Occupancy detection from electricity consumption data," in *Int. Workshop on Embedded Systems For Energy-Efficient Buildings, BuildSys13*, New York, NY, USA, 2013, ACM, pp. 101–108.

[5] D. Chen, S. Barker, A. Subbaswamy, D. Irwin, and P. Shenoy, "Nonintrusive occupancy monitoring using smart meters," in *Int. Workshop on Embedded Systems For Energy-Efficient Buildings, BuildSys13*, New York, NY, USA, 2013, ACM, pp. 91–98.

[6] S. N. Patel, T. Robertson, J. A. Kientz, M. S. Reynolds, and G. D. Abowd, "At the flick of a switch: Detecting and classifying unique electrical events on the residential power line," in *Int. Conf. on Ubiquitous Computing, UbiComp07*, Berlin, Germany, 2007, Springer-Verlag, pp. 271–288.

[7] A. G. Ruzzelli, C. Nicolas, A. Schoofs, and G. M. P. OHare, "Realtime recognition and profiling of appliances through a single electricity sensor," in *Int. Conf. on Sensor Mesh and Ad Hoc Communications and Networks, SECON10*, Boston, Massachusetts, USA, 2010, IEEE ComSoc, pp. 1–9.

[8] N. A. M. Rosdi, F. H. Nordin, and A. K. Ramasamy, "Identification of electrical appliances using non-intrusive magnetic field and probabilistic neural network (pnn)," in *Int. Conf. on Power and Energy, PECON14*, Kuching, Sarawak, Malaysia, 2014, IEEE, pp. 47–52.

[9] N. Klingensmith, D. Willis, and S. Banerjee, "Extracting events from spatial time series," in *nt. Workshop on Embedded Systems For Energy-Efficient Buildings,BuildSys13*, New York, NY, USA, 2013, ACM, pp. 5–8.

[10] Robert Ghrist, "Barcodes: The persistent topology of data," *Bulletin (New Series) of the American Mathematical Society*, vol. 45, pp. 61–75, jan 2008.

[11] Allen Hatcher, *Algebraic Topology*, Cornell University, New York, 2001.

[12] H. Edelsbrunner and J. Harer, "Persistent homology - a survey," *Contemporary Mathematics*, vol. 453, pp. 257–282, 2008.

[13] Afra Zomorodian and Gunnar Carlsson, "Computing persistent homology," *Discrete & Computational Geometry*, vol. 33, no. 2, pp. 249–274, 2005.

[14] Nina Otter, Mason A. Porter, Ulrike Tillmann, Peter Grindrod, and Heather A. Harrington, "A roadmap for the computation of persistent homology," 2017, preprint on arXiv at https://arxiv.org/abs/1506.08903.

[15] S. Biasotti, B. Falcidieno, D. Giorgi, and M. Spagnuolo, *Mathematical tools for shape analysis and description*, vol. 6 of *Synth. Lect. Comput. Graph. Animat.*, 2014.

[16] Natasa Jonoska and Masahico Saito, *Discrete and Topological Models in Molecular Biology*, Natural Computing Series. Springer Berlin Heidelberg, 2014.

[17] V. De Silva and R. Ghrist, "Coverage in sensor networks via persistent homology," *Algebraic and Geometric Topology*, vol. 7, pp. 339358, 2007.

[18] Gunnar Carlsson, Tigran Ishkhanov, Vin de Silva, and Afra Zomorodian, "On the local behaviour of spaces of natural images," *International Journal of Computer Vision*, vol. 76, pp. 1–12, 2008.

[19] Jose A. Perea and John Harer, "Sliding windows and persistence: An application of topological methods to signal analysis," *Foundations of Computational Mathematics*, vol. 15, no. 3, pp. 799–838, 2015.

[20] Cássio M.M. Pereira and Rodrigo F. de Mello, "Persistent homology for time series and spatial data clustering," *Expert Syst. Appl.*, vol. 42, no. 15, pp. 6026–6038, sep 2015.

[21] S. Biasotti, L. De Floriani, B. Falcidieno, P. Frosini, D. Giorgi, C. Landi, L. Papaleo, and M. Spagnuolo, "Describing shapes by geometrical-topological properties of real functions," *ACM Comput. Surv.*, vol. 40, no. 4, pp. 12:1–12:87, 2008.

[22] Andrew Tausz, Mikael Vejdemo-Johansson, and Henry Adams, "JavaPlex: A research software package for persistent (co)homology," in *Proceedings of ICMS 2014*,

Han Hong and Chee Yap, Eds., 2014, Lecture Notes in Computer Science 8592, pp. 129–136, Software available at http://appliedtopology.github.io/javaplex/.