

# Topological Data Analysis – group project

---

## Sensors

---

Nejc Kišek, Žan Klaneček

June 8, 2018

## 1 Introduction

In this project we are trying to find optimal parameters for a sensor network which covers the whole surface of the earth. The network is determined by two parameters:

- $r$ , which is the distance over which the sensors can communicate with each other – each sensor can communicate with other, if they are at most  $r$  away.
- $R$ , which is the radius of the surrounding area in shape of a circle where the sensor can gather data.

Given the coordinates of each sensor, we had to determine the lowest possible parameters  $r$  and  $R$  so that:

- the sensor network is connected,
- the sensor network covers the whole sphere (surface of the Earth).

Furthermore, once the parameters  $r$  and  $R$  are established, obsolete sensors should be removed. These are sensors whose removal would not change the connectivity and coverage of the sensor network. Lastly, a data generator that distributes  $n$  points evenly on the surface of a sphere will be described. With this generator, parameters  $r$  and  $R$  of the sensor network will be as small as possible.

## 2 Topological solution

In this section theoretical background for our problem is described. Firstly, if we want to achieve sensor network connectivity (all sensors are connected), we should have a look at Vietoris-Rips complex. The properties of Vietoris-Rips are described with following relations:

- sensors:  $S_i = (r_i, \phi_i, \theta_i)$ ,
- sensor connections  $\{S_i, S_j\} \subset S; d(S_i, S_j) \leq r$ ,
- $F \subset S$  is a simplex in  $VR_r(S)$ , if  $\text{diam } F \leq r$ ,

where  $r$  is the parameter of the Vietoris-Rips complex and  $d(S_i, S_j)$  is Euclidean distance between two sensors. We can see that two vertices in the Vietoris-Rips complex will be connected if they are at most  $r$  away, which is also the requirement for the network to communicate.

To get a connected sensor network, such  $r$  (as small as possible) should be determined, that Vietoris-Rips graph on a given distribution of sensors is connected (i.e. has one component).

Secondly, if we want to achieve coverage of the entire sphere, Čech complex will come in handy:

- sensors:  $S_i = (r_i, \phi_i, \theta_i)$ ,
- $B_R(x)$  closed ball with radius  $R$  around  $x$ ,
- $C_R = \{\sigma \subset S, \cap_{x \in \sigma} B_R(x) \neq \emptyset\}$ .

Three vertices in a Čech complex form a triangle if the balls with radius  $R$  around them have a common intersection. From that follows that every triangle in the Čech complex represents an area that is completely covered by the sensors. This means that the Čech complex on our ideal sensor network covers the whole surface with triangles and is therefore homeomorphic to a sphere.

In practice, we ignored higher dimensional vertices, because they only appear when more than three sensors are very close together, which is not an issue. We therefore just looked at the first 2 Betti numbers:  $\beta_0$  = number of components, which should be one and  $\beta_1$  = number of cycles/holes in the complex, which should be zero.

### 3 Connectivity

To determine if the sensor network is connected, we computed Vietoris-Rips complexes with different  $r$  parameters and observed the graphs represented by points and edges of such complexes. We were looking for a graph with just 1 component.

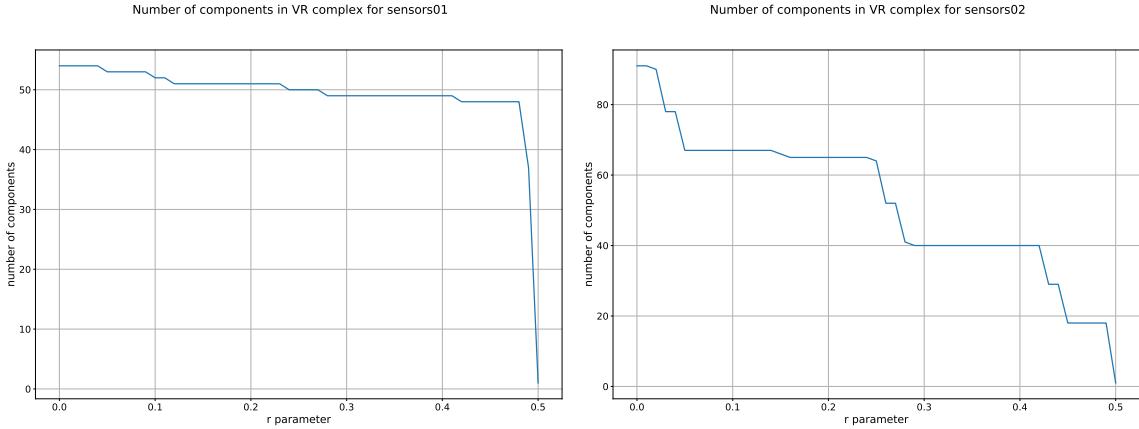


Figure 1: Number of components as we increase the  $r$  parameter

In Figure 1 we can see how the number of components changes as we increase the  $r$ . In the first dataset, the points were evenly distributed on the sphere, so most of them had roughly equal spacing between them – we can see that the number of components decreases slowly at first, and quickly drops to one when we approach 0.5. In the second dataset, points around the equator were further apart than those around the poles – we can see that the number of components gradually decreases.

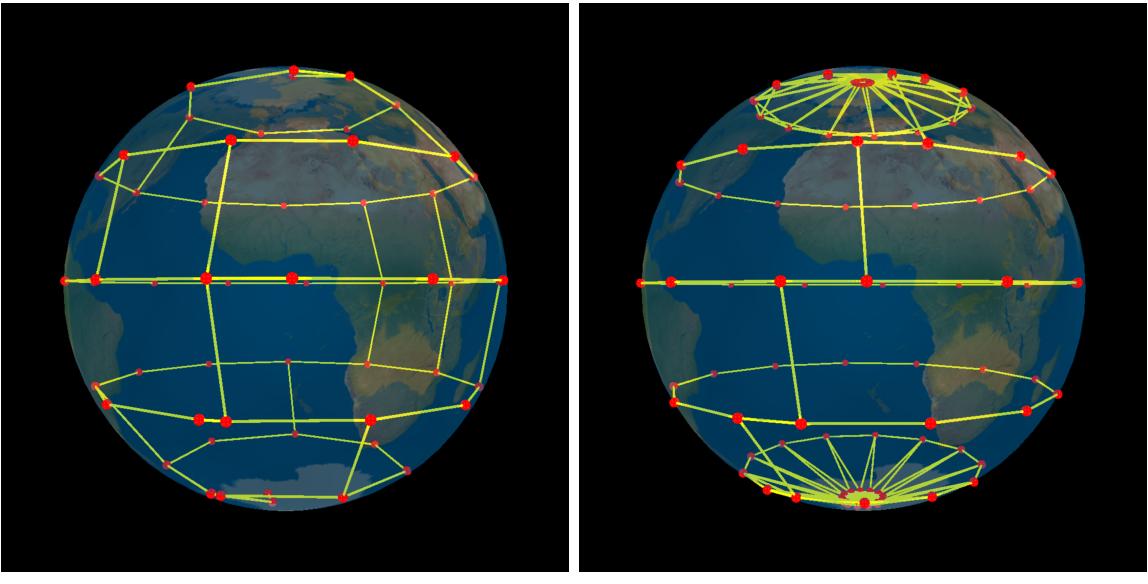


Figure 2: Connections between sensors

In Figure 2 we can see the edges and vertices of a Vietoris-Rips complex that represent the communication between sensors. On the right picture we can also see the points around the poles, which we mentioned before, and which have much shorter distances and are therefore also connected when  $r$  is smaller.

## 4 Coverage

To determine the coverage of our sensor network we computed multiple Čech complexes with multiple  $R$  parameters and looked at their first two Betti numbers.

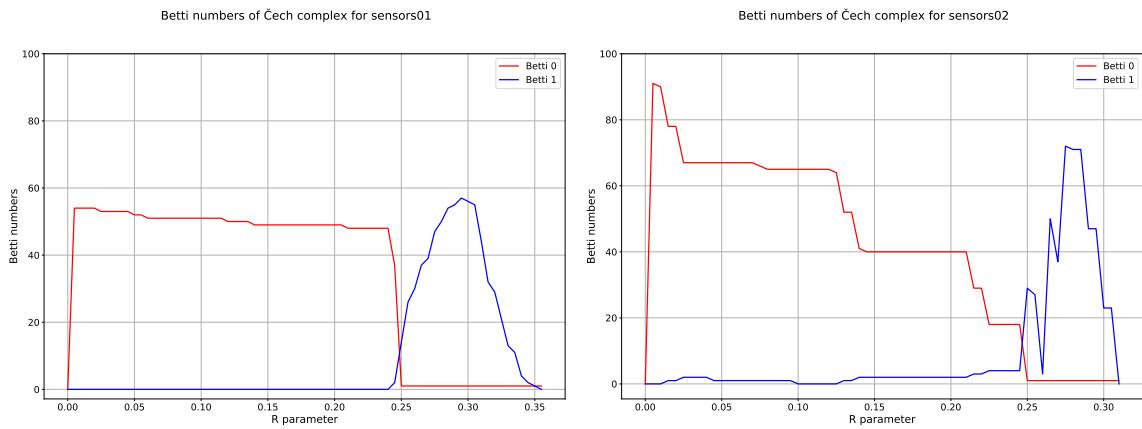


Figure 3:  $\beta_0$  and  $\beta_1$  as we increase the  $R$  parameter

In Figure 3 can see, that  $\beta_0$ , the number of components, represented by the red line has the same shape as on the previous Vietoris-Rips plots. We can also see that the blue line, the number of cycles in the complex, is low at first, rises when the points are connected together and then decreases when the cycles turn into triangles.

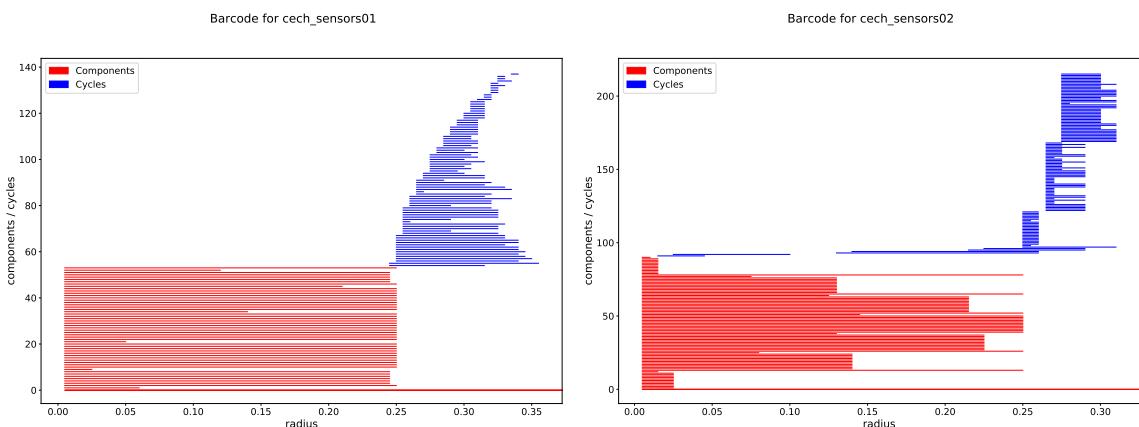


Figure 4: Barcodes for the Čech complex

Barcodes in Figure 4 show us a very similar pattern.

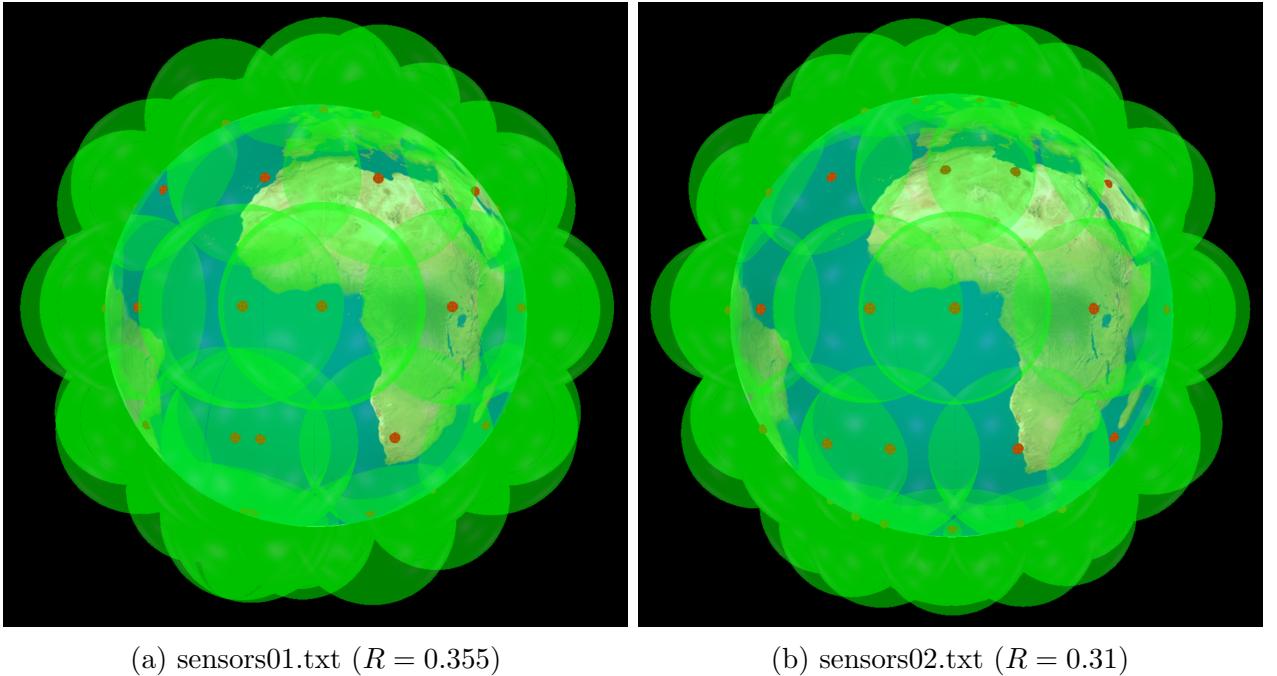


Figure 5: Connections between sensors

In Figure 5 we can see the coverage of each sensor, represented by a green ball with radius  $R$  around it. We can see that there is some overlap between them (especially around the poles of the second dataset), but we can also see, that decreasing the  $R$  would create holes in between some of the balls.

## 5 Redundant sensors

Once the parameters of the sensor network  $r$  and  $R$  were established, we find any redundant sensors. We first tried to go through all possible combinations ( $n!$ ), which turned out to be very time demanding, so we reduced the complexity with the following simple algorithm:

- We ignored the cut vertices of the graph on the VR complex. Removing them would separate the network in multiple components.
- We ignored all sensors with less than 3 neighbors in radius  $R$  around them. Removing them would create a hole in the Čech complex.
- We then randomly selected points from the remaining list and removed them from the network if it was safe to do so (we checked if the remaining network would still satisfy the requirements).

The best solution that we got by randomly picking points from the complex was further improved by going through all the possible combinations of remaining sensors that could be safely removed. This was now faster, since we already removed

most of the vertices that could be removed in the previous step. In the end, our algorithm for optimizing the sensor network was a compromise between finding a good solution and computation time.

## 6 Data generator

Lastly, let's have a look at the data generator that distributes  $n$  points evenly on the surface of a sphere. In order to meet this requirement we should look on points as electrons that are bound on the surface (i.e. in spherical coordinates have fixed radius). The natural state would be the state with the lowest electrostatic-potential energy  $V$  (eq. (2)). This is due to the fact that electrons repel from each other according to Coulomb's inverse square law (eq. (1))

$$|\mathbf{F}_{ij}| = k_e \frac{e^2}{|\mathbf{r}_i - \mathbf{r}_j|^2}, \quad (1)$$

$$V = \sum_{i \neq j} V_{ij} \propto \sum_{i \neq j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}, \quad (2)$$

where  $k_e$  is Coulomb's constant,  $e$  charge of electron and  $\mathbf{r}_i$  local vector for  $i-th$  point. It should be emphasized that potential energy was determined as sum of inverse Euclidean distances between all points.

### 6.1 Simulated annealing

In order to determine the minimum of electrostatic-potential energy (eq. (2)) we used algorithm for simulated annealing with some Monte-Carlo modifications. At the beginning we randomly distributed  $n$  points on the surface of a sphere with radius  $r$  and selected the initial temperature of the system  $T$ . In each iteration a random point was chosen (one point was fixed, so  $n - 1$  available) and moved it according to Gaussian distribution with appropriate parameters ( $\mu = 0$  and small enough  $\sigma$ ). After that, the difference in energy was calculated  $\Delta E$ . If the energy of newly distributed points was less than before (i.e.  $\Delta E < 0$ ) the move was accepted. Otherwise ( $\Delta E \geq 0$ ) the move was accepted with the probability  $\exp(-\frac{\Delta E}{T})$ . The higher the difference in energy the lower are chances that we accept the change. Similar for the temperature, the lower the temperature the lower is the probability for accepting a bad move. If for a certain temperature  $T$  enough changes  $N_T$  were accepted, we decrease the temperature  $T$  for  $\Delta T$ . Continue with another (or same) random point.

### 6.2 Generated datasets

In this report we will include two generated datasets, for  $n = 4$  and for  $n = 50$ . We can see that in case of four points they distribute themselves in tetrahedral geometry and for  $n = 50$  we get nice homogeneous even distribution of points.

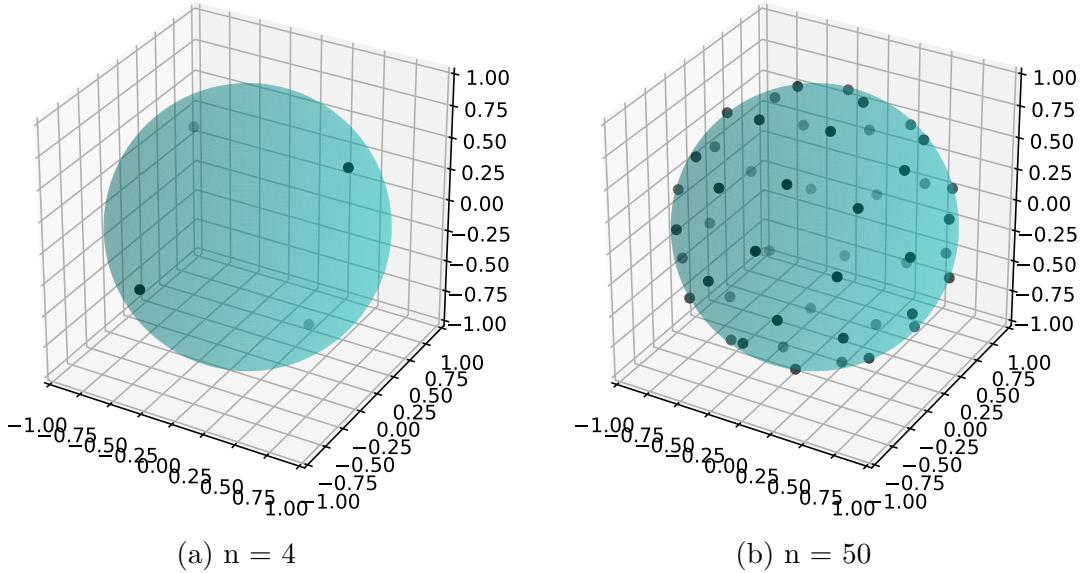


Figure 6: Even distribution of generated  $n$  points on the surface of a sphere.

7 Conclusion

Topological data analysis is well suited for solving the problem of finding connections and coverage of a sensor network, since the definitions of connections in a graph and coverage of a sensor are essentially the same as definitions of edges in Vietoris-Rips and triangles in Čech complexes. The algorithms for computing the complexes are also fast enough that computing the solutions on the provided datasets takes just a few seconds.

An area that could be improved is the algorithm for finding redundant sensors, where we didn't come up with any topological way of solving the problem. An idea that we thought about but didn't explore in detail was to find simplices of a Čech complex with very high dimensions and removing sensors from them.

## References

- [1] Vietoris-Rips. [https://en.wikipedia.org/wiki/Vietoris\\_Rips\\_complex](https://en.wikipedia.org/wiki/Vietoris_Rips_complex) (5.6.2018).
  - [2] Čech-complex. [https://en.wikipedia.org/wiki/Cech\\_complex](https://en.wikipedia.org/wiki/Cech_complex) (5.6.2018).
  - [3] Dionysus. <http://www.mrzv.org/software/dionysus/> (5.6.2018).
  - [4] Miniball algorithm. <https://github.com/weddige/miniball> (5.6.2018).
  - [5] Vpython. <http://vpython.org/> (5.6.2018).

[6] Lecture notes from prof. dr. Neža Mramor Kosta.