

STALKING YOURSELF

S.1
**DESIGNING
ASSOCIATIVITY**

**Luis
Fraguada**

**Rodrigo
Aguirre**

**Assignment 2
Geolocation #2**

Matteo Silverio

INDEX

RECORDING THE GPS DATA

PROCESSING THE GPS DATA (*.KML)

IMPORTING THE GPS DATA INTO GRASSHOPPER

DISPLAYING THE DATA

Introduction:

The following description would be a tutorial that explains the steps you need to process GPS data using Rhinoceros 3D. To do that, you should have:

- Smartphone with a GPS tracking application (RunKeeper)
- Internet connection
- Notepad++
- Google Earth
- McNeel Rhinoceros 3D
- Grasshopper
- gHowl (grasshopper plugin)
- LunchBox (grasshopper plugin)

I. RECORDING THE GPS DATA

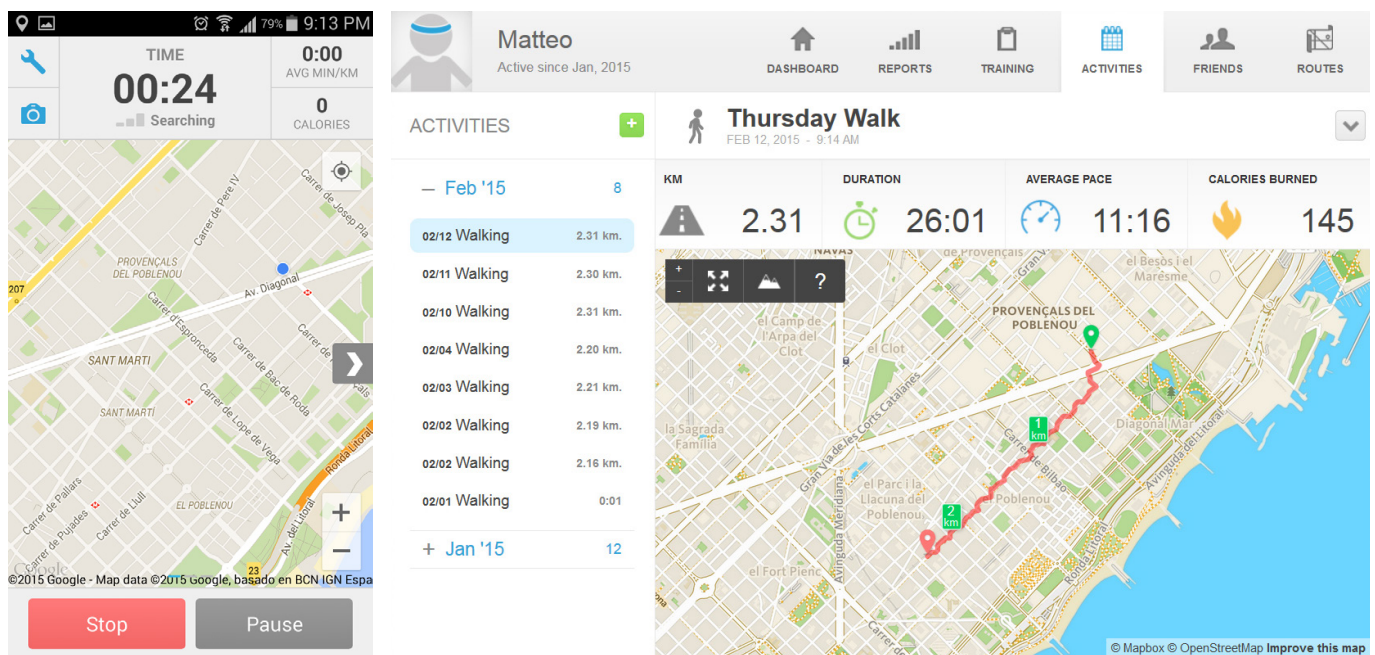
Smartphones allow people to do many activities, included know where they are thanks to their GPS antenna. There are many apps for smartphones that track our movements and most of these have been developed to check outdoor fitness activities.

However, the stored data will be also useful to make statistic researches as well as to study people movements. In this tutorial we will try to process these GPS data in order to understand how many time the "test subject" (let's call him Matteo) has passed in a specific area of Barcelona.

Method:

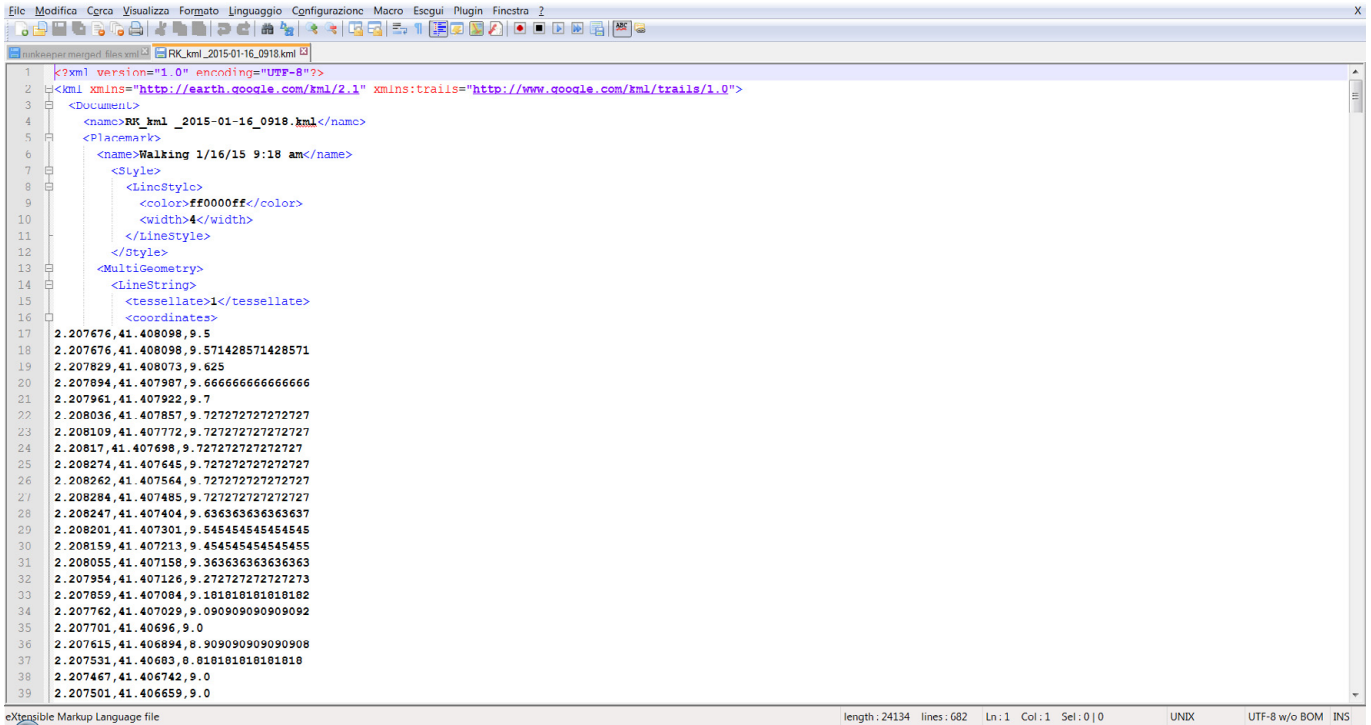
First of all, recording his movements using a GPS tracking app. For this experiment RunKeeper for Android has been used. RunKeeper is a free app available for both Google and Apple devices and has been developed to check outdoor fitness activities.

All the activities could be saved, so each user can check his data from the official **RunKeeper** webpage. The files can be downloaded as .gpx or .kml format.



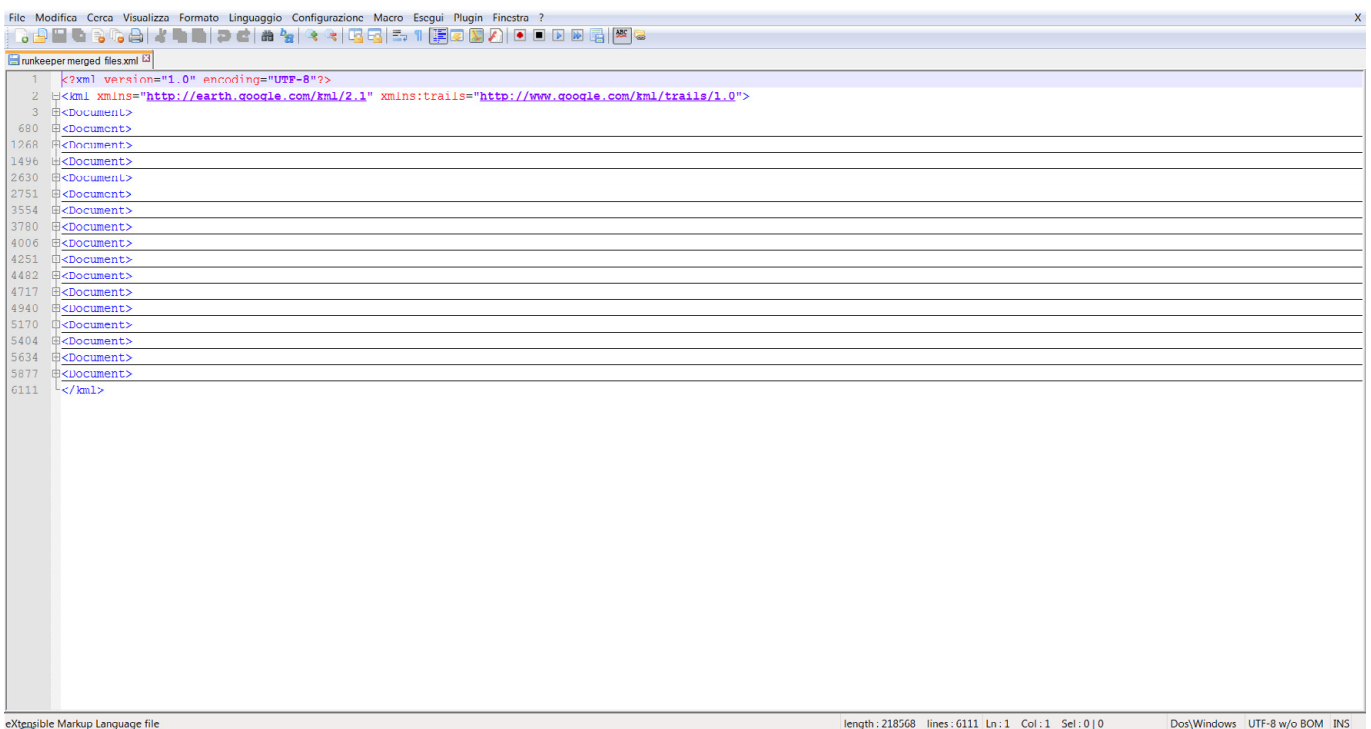
II. PROCESSING THE GPS DATA (*.KML)

Downloading all the stored data as **kml format** (Keyhole Markup Language). After that, it could be useful to join all the files together to speed the loading data process into Grasshopper. Therefore opening **Notepad++** and merging all the files together.



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <kml xmlns="http://earth.google.com/kml/2.1" xmlns:trais="http://www.google.com/kml/trais/1.0">
3 <Document>
4 <name>RK_kml_2015-01-16_0918.kml</name>
5 <Placemark>
6 <name>Walking 1/16/15 9:18 am</name>
7 <Style>
8 <LineStyle>
9 <color>ff0000ff</color>
10 <width>4</width>
11 </LineStyle>
12 </Style>
13 <MultiGeometry>
14 <LineString>
15 <tessellate>1</tessellate>
16 <coordinates>
17 2.207676,41.408098,9.5
18 2.207676,41.408098,9.571428571428571
19 2.207829,41.408073,9.625
20 2.207894,41.407987,9.666666666666666
21 2.207961,41.407922,9.7
22 2.208036,41.407857,9.727272727272727
23 2.208109,41.407772,9.727272727272727
24 2.20817,41.407698,9.727272727272727
25 2.208274,41.407645,9.727272727272727
26 2.208262,41.407564,9.727272727272727
27 2.208284,41.407485,9.727272727272727
28 2.208247,41.407404,9.636363636363637
29 2.208201,41.407301,9.545454545454545
30 2.208159,41.407213,9.454545454545455
31 2.208055,41.407158,9.363636363636363
32 2.207954,41.407126,9.272727272727273
33 2.207859,41.407084,9.181818181818182
34 2.207762,41.407029,9.090909090909092
35 2.207701,41.40696,9.0
36 2.207615,41.406894,8.909090909090908
37 2.207531,41.40683,9.818181818181818
38 2.207467,41.406742,9.0
39 2.207501,41.406659,9.0
length: 24134 lines: 682 Ln: 1 Col: 1 Sel: 0|0 UNDX UTF-8 w/o BOM INS
```

Notepad++ recognizes several languages (included kml), hence, analysing the files their code structures can be understood. The goal of this analysis is understanding the file hierarchy and merging all the collected data in a smart way.



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <kml xmlns="http://earth.google.com/kml/2.1" xmlns:trais="http://www.google.com/kml/trais/1.0">
3 <Document>
680 <Document>
1268 <Document>
1496 <Document>
2630 <Document>
2751 <Document>
3554 <Document>
3780 <Document>
4006 <Document>
4251 <Document>
4482 <Document>
4717 <Document>
4940 <Document>
5170 <Document>
5404 <Document>
5634 <Document>
5877 <Document>
6111 </kml>
length: 218568 lines: 6111 Ln: 1 Col: 1 Sel: 0|0 Dos/Windows UTF-8 w/o BOM INS
```

III. IMPORTING THE GPS DATA INTO GRASSHOPPER

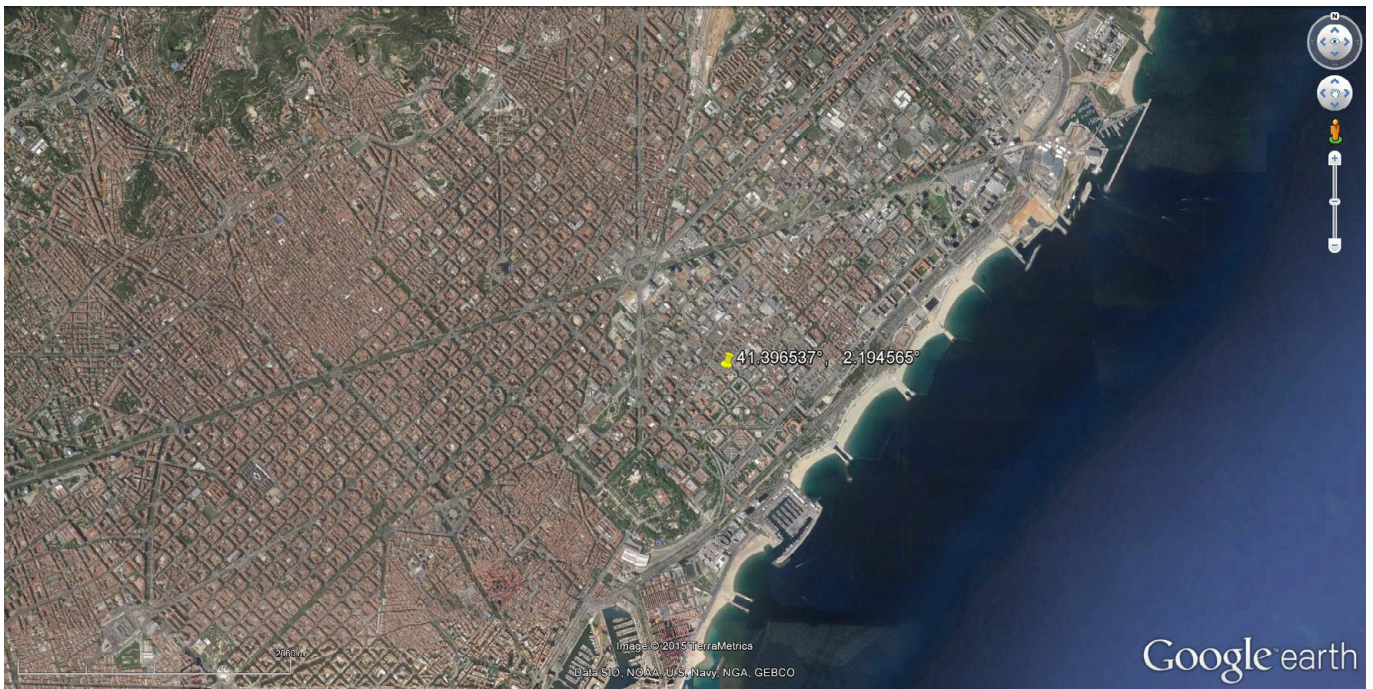
Using **gHowl** commands to import the merged GPS file into Rhino.

Opening Grasshopper, loading the .xml file using the **File Path Command** and connecting it to the **XML Parser Component**. This component can read .kml files and recognize their contents. In this file, all the GPS coordinates are contained after the **<coordinates> Instruction**, so all the branches called "coordinates" should be extracted.

Therefore, using the **Expression** and the **Cull Pattern Commands**, extracting the "coordinates" branches. In the same way extracting also the coordinate dates, so the branches can be related to their specific time period.

Finally, using the **Text Split Command** to divide all the data within the same branch. The result should be a series of cloud points contained into a certain amount of branches (one branch = one path) and an another list with all the branch references (dates).

The next step should be displaying the point into a scaled map so each movement can be recognized. Opening **Google Earth**, pinning a location (saving its coordinates) and saving the resulted image.



Opening the image into Rhino, scaling it and importing the pinned point into Grasshopper. Using the **XYZ to Geo Command** translating the point into a Geo coordinate, then using the **Geo to XYZ Command**, positioning all the .xml cloud points on the imported image.

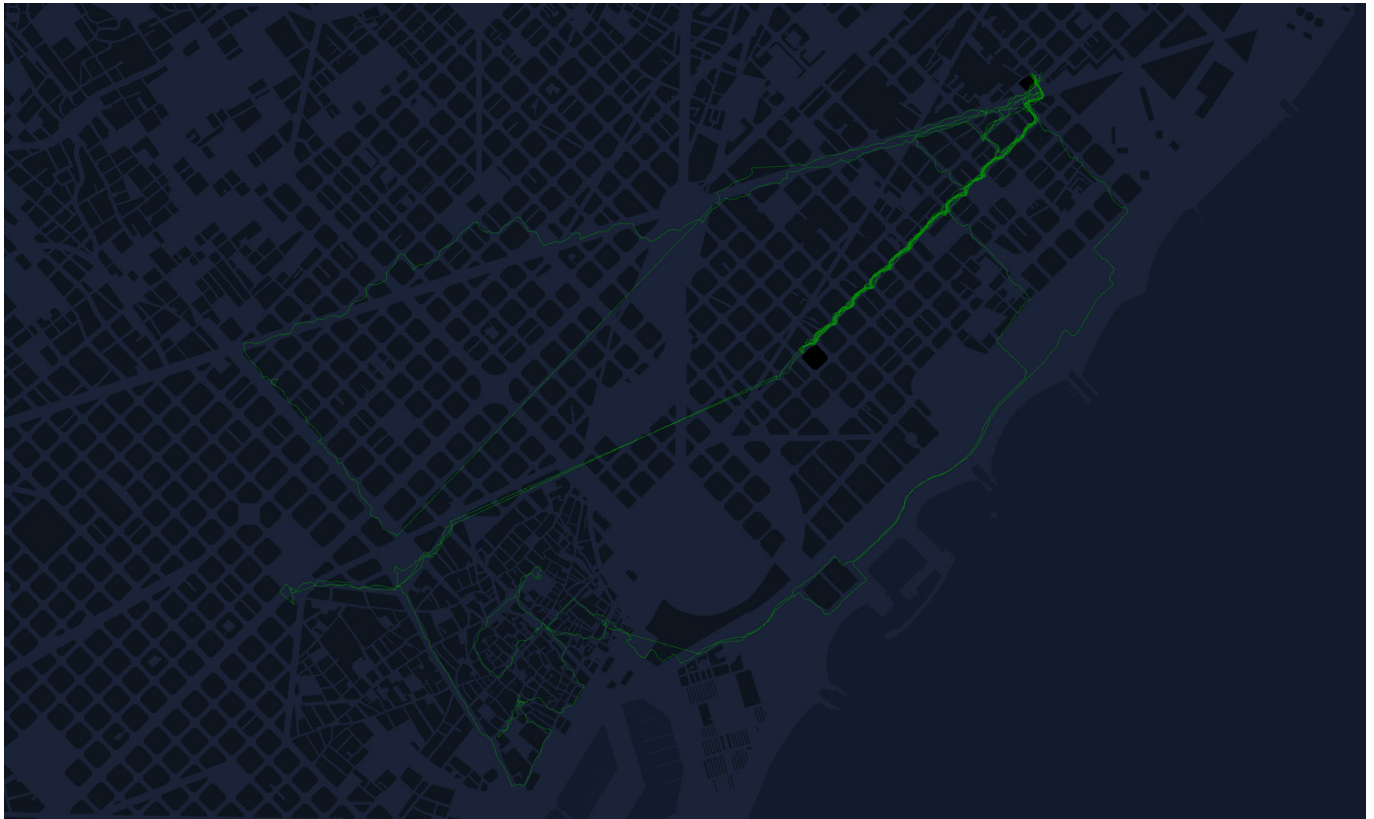


IV. DISPLAYING THE DATA

All the tracked movements are now displayed. Let's try to show these data in a more efficient and interesting way. First of all the base image should be improved. Thus a Barcelona dwg file has been imported and modified, obtaining a cleaned city map.



Using the Rhino **Hatch Command** all the main urban elements (buildings, streets, ...) have been "painted" in blue tones, reaching a nicer base for our charts.



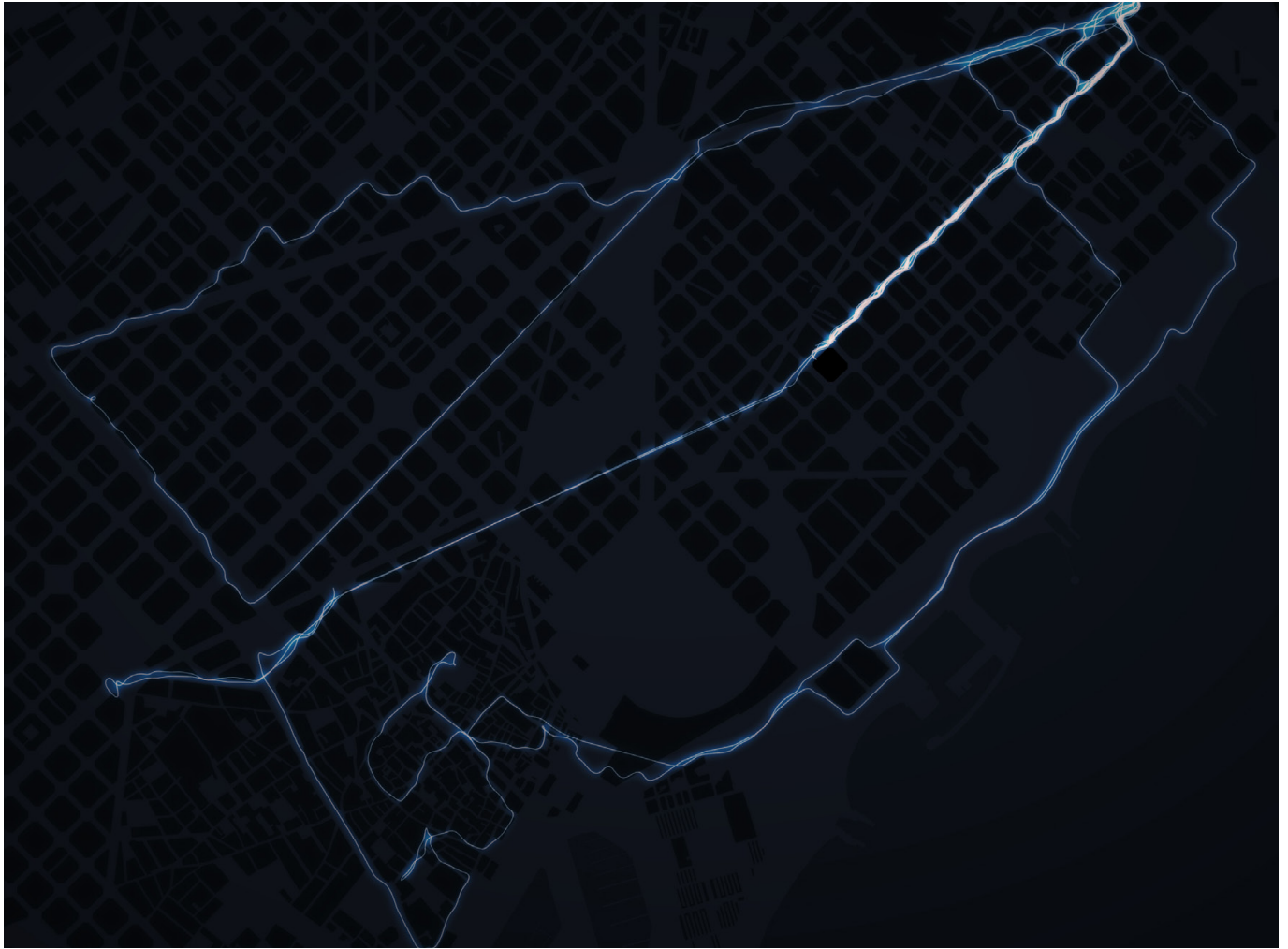
Normal paths:

Using the Interpolate Command, drawing the path lines. The lines are floating to the 3D space (Barcelona is not a perfectly flat City), so using the **Project Command** all the lines have been flattened to the plan. Moreover, using the **Smooth Polyline Command** all these lines have been rounded.

Creating **Mesh pipe objects** throughout the polylines in order to highlight them. In addition to that, these surfaces have been painted in white (in contrast with the dark blue background) using the **Colour RGB Command** that also controls the alpha channel (transparency).

Thanks to this feature, the objects are a bit transparent and their overlapping empathize the more covered paths.





How many time did he pass in a certain area?

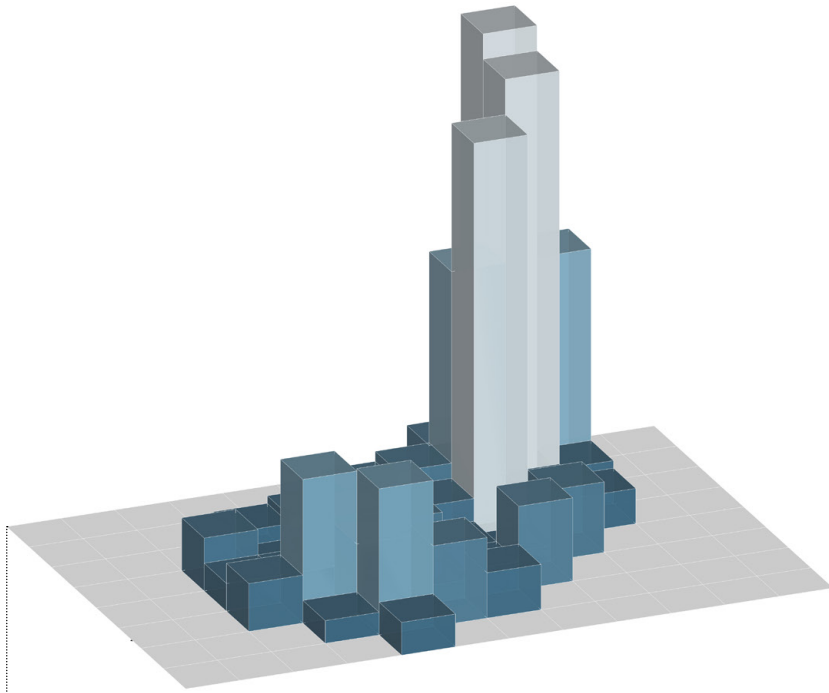
To answer to this question, the City has been divided in 96 equal parts (it should be also divided in neighbourhood or in more specific areas, this division has been set just to speed up the development).

Then, using the **Divide Length Command** each line has been divided using 50 as parameter (50 = one point every 50 meters). Thanks to this solution, the diagram can consider also the metro trips.

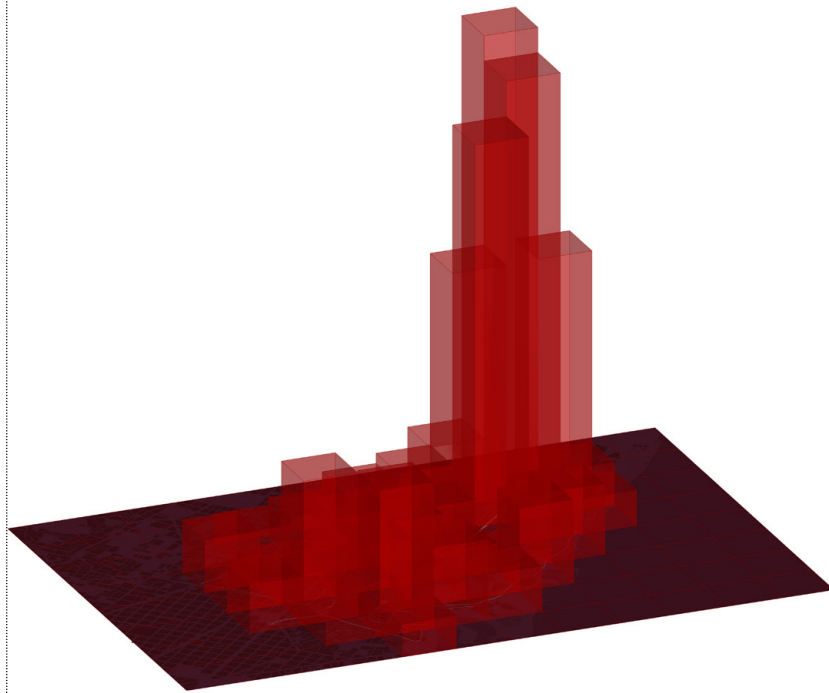
Using the **Point in Curves**, the **Cull Pattern** and the **List Length Commands** is possible to calculate how many time Matteo spent in each area. After that, rescaling all the data and extruding all the areas according to the obtained value.

Note:

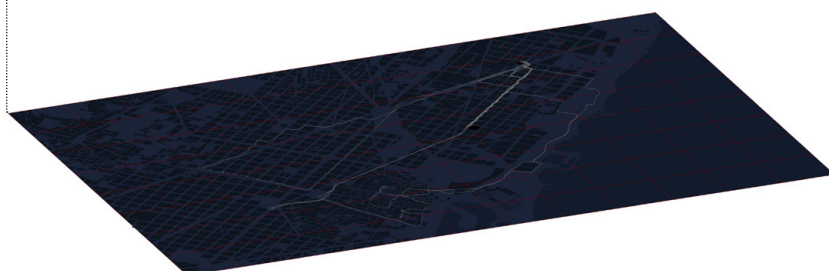
The diagram doesn't show the real time spent in each city cluster, because I can't understand the relation between GPS points and time. Moreover there is no signal during metro trips. Probably is more exact asserting that the diagram shows how many meters the users has walked (or remained) inside each cluster.



EXTRUDED CLUSTERS COLORED
ACCORDING TO THEIR HEIGHT

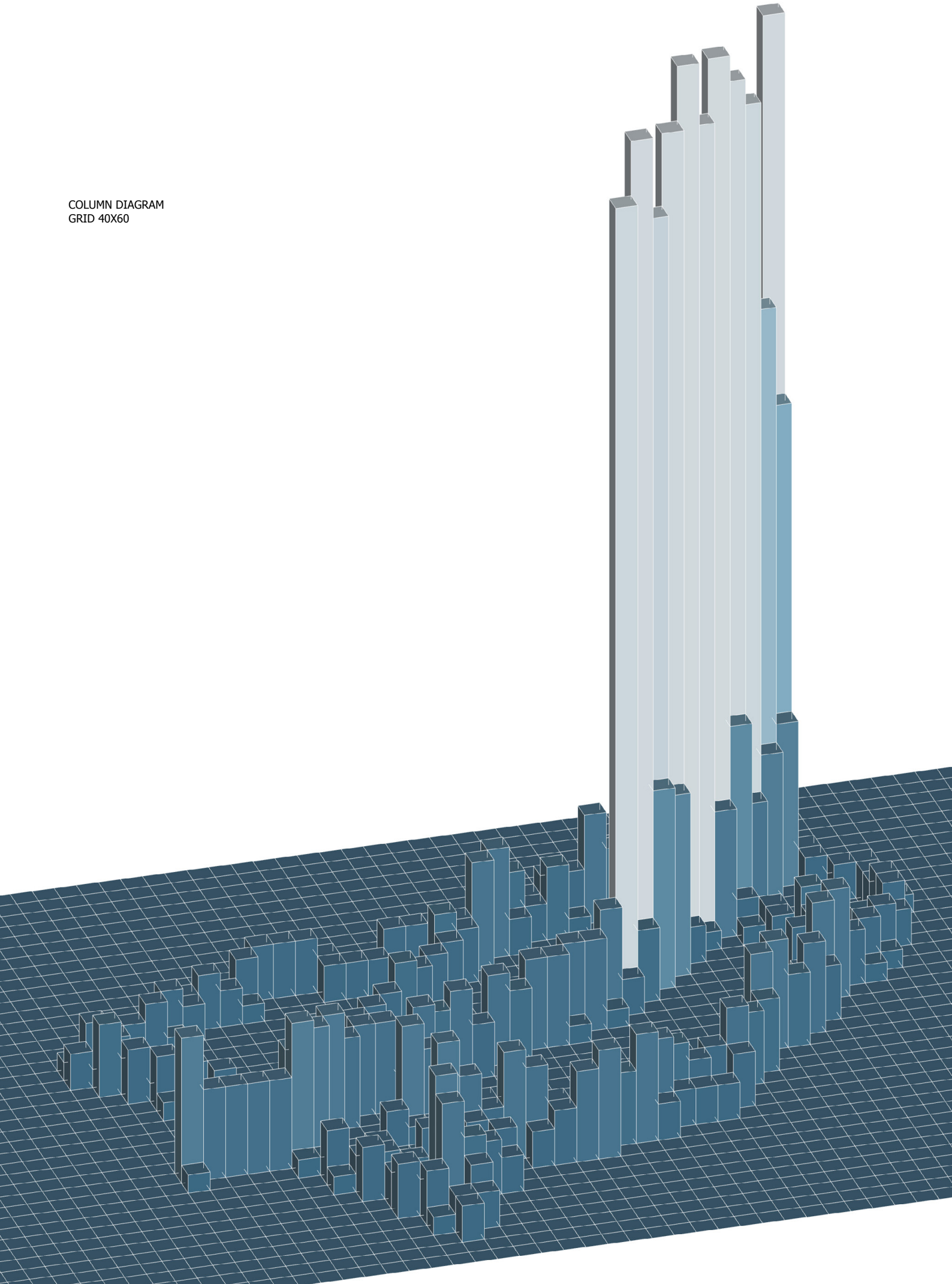


CLUSTERS EXTRUDED ACCORDING
TO THE DATA



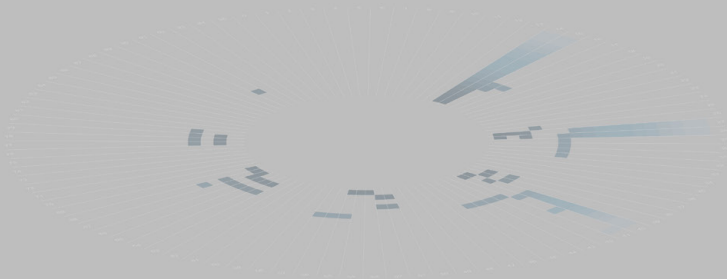
PLAN DIVIDED IN 96 PARTS
(LUNCHBOX QUAD PANELS 8x12)

COLUMN DIAGRAM
GRID 40X60

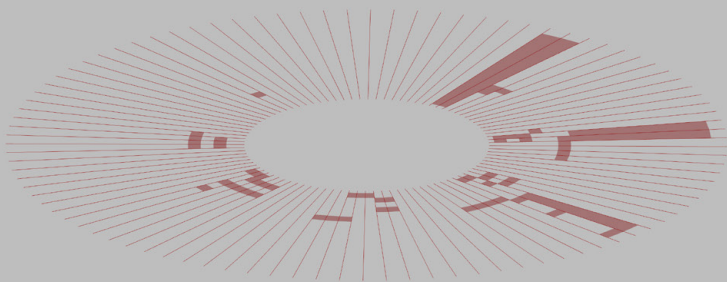


Final 2D chart:

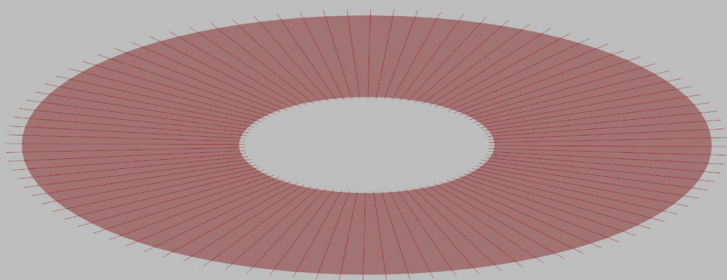
A 2D chart could be useful to understand more in deep the movements across the City. This new chart should show the test subject movements in each recorded day, trying to understand his behaviour as well as his daily/weekly habits, studying particular custom changes during weekends.



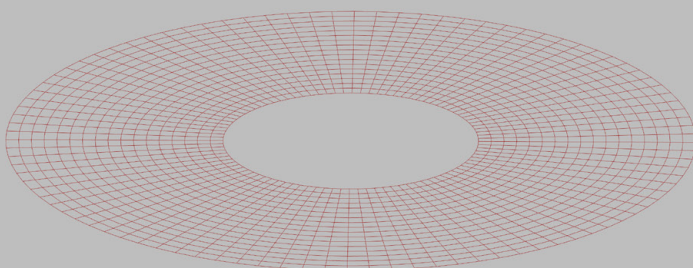
CELLS SORTED BY COLORS



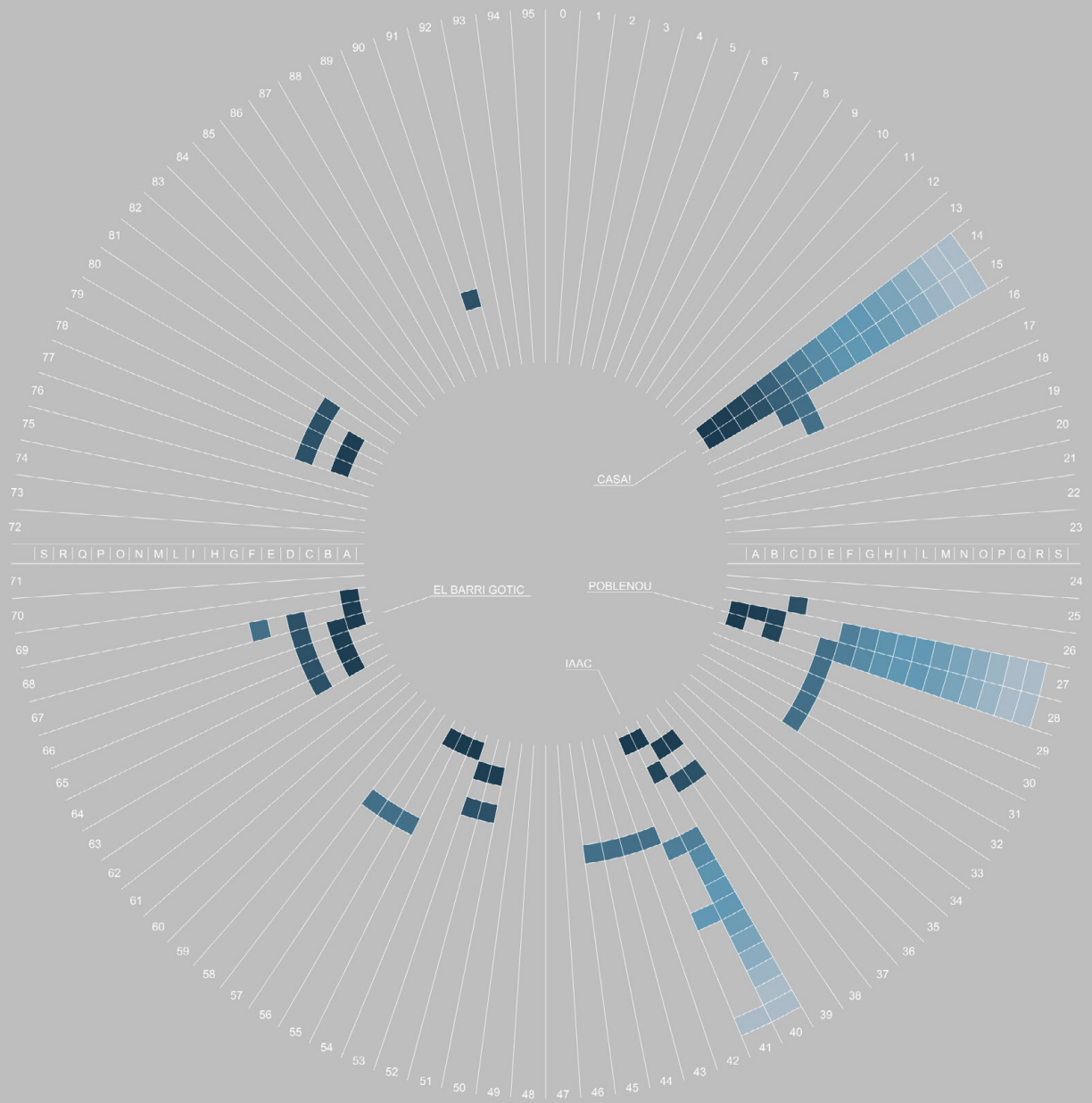
INTERPOLATION
CITY CLUSTER/TRAKKED PATH



CELLS FILLED WITH MESHES



BASIC RADIAL GRID



11	10	9	8	7	6	5	4	3	2	1	0
23	22	21	20	19	18	17	16	15	14	13	12
35	34	33	32	31	30	29	28	27	26	25	24
47	46	45	44	43	42	41	40	39	38	37	36
59	58	57	56	55	54	53	52	51	50	49	48
71	70	69	68	67	66	65	64	63	62	61	60
83	82	81	80	79	78	77	76	75	74	73	72
95	94	93	92	91	90	89	88	87	86	85	84

- A 1/16/15 9:18 am
- B 1/18/15 10:42 am
- C 1/19/15 9:17 am
- D 1/23/15 10:42 am
- E 1/24/15 7:10 pm
- F 1/25/15 4:53 pm
- G 1/27/15 9:30 am
- H 1/27/15 8:54 pm
- I 1/28/15 9:17 am
- L 1/29/15 9:13 am
- M 1/30/15 9:14 am
- N 2/2/15 8:54 am
- O 2/2/15 6:02 pm
- P 2/3/15 9:11 am
- Q 2/4/15 9:17 am
- R 2/10/15 9:07 am
- S 2/11/15 9:42 am