The Walking Data

A pedestrian simulation visualization

By Nicholas Molyneaux, Semin Kwak, and Gael Lederrey

Introduction

Being able to visualize data is critical in many scientific applications. Nicholas Molyneaux is currently doing his PhD in the field of pedestrian modelling and simulation. The objective is to investigate control strategies for pedestrian traffic. As no readily available free software exists for visualizing pedestrian tracking data, we decided to create a web-based tool for this. The goal is to create an interactive visualization where one can explore the movements of the pedestrians inside the simulated environment and also get some simple statistics about the data.

This visualization has been developed using Chrome as a browser. We have paid attention that the functionalities also work with firefox, but it is possible that some minor bugs are present with firefox.

Target audience

Naturally, given the motivation of the project one of the key users will be Nicholas as this will be directly used for visualizing the results from his pedestrian simulations. Nevertheless, there are other audiences to whom this viz can be useful. In the context of the european project TRANS-FORM (http://www.trans-form-project.org/), other partners can be interested in visualizing the results from the pedestrian simulations. These are people from a scientific community or other technical jobs. The 3D part which is less technical but more fun is targeted at a less technical audience.

Idea of the visualization

The main visualization of walking pedestrians consists of seeing them move in a 2D space. An example of a previous visualization (video) is given below.

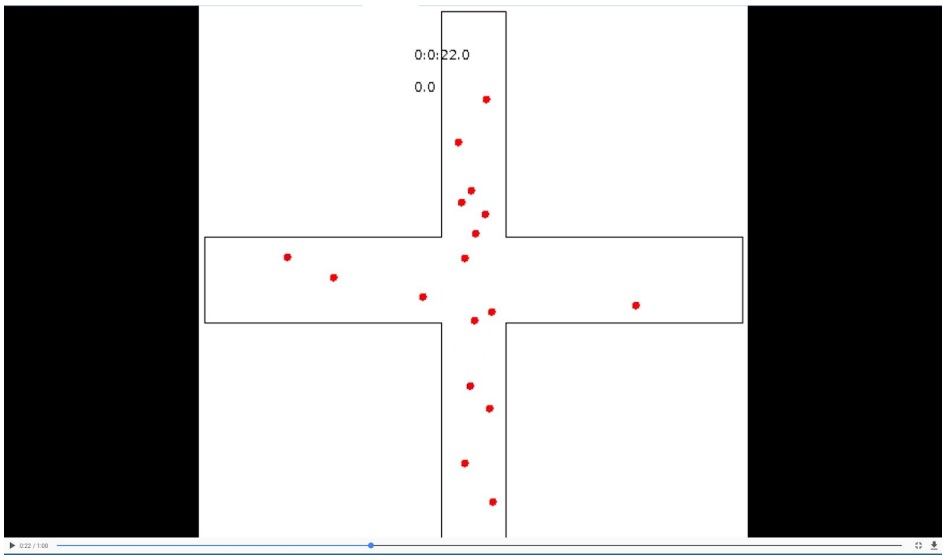


Figure 1: Previous visualization of Nicholas Molyneaux simulator done in Java.

The first idea for our project was to reproduce such a visualization using d3.js. This visualization will be the main the visualization of our project. Everything we do later on will be based on enhancing this visualization. Thus, we had to think about the improvement we want to make research-wise. Here are the ideas we want to implement:

- Pedestrians moving around on top of background map
  - Color pedestrian by current
    - Velocity
    - Density
    - Difference with desired velocity
- Overlay over background map
  - Flows (thick arrows going from O to D)
  - Heatmap: places where there are lots of pedestrians
  - Temporal voronoi diagrams (inside sub-area as well)
- Optional
  - Show arrival/departure vehicles
  - Show state of control devices (gates, moving walls)

In terms of research, we also need to provide some static graphs to show the results. We, thus, decided to implement the following graphs for the statistical analysis:

- OD (Origin-Destination) graphs as Edge bundling.
  - We want to be able to select some ODs by clicking on them on the 2D viz.
- Plot quantities (mostly histograms) such as
  - Velocity
  - Density
  - Travel time
  - Travel distance

For all of these statistics, we need to be able to select the data range in terms of time and space, *i.e.* by group of ODs..

Finally, we decided that we wanted to create a 3D visualization of the pedestrians walking around the structure for the fun. We won't add any of the overlays available for the 2D viz. The different steps for this last visualization are the following:

- Building 3D models of Lego characters on Blender
- Animate them for the walking part.
- Use the library Three.js to integrate the models and the structures.

This last visualization may not be delivered at the end of the project since it is not relevant in terms of research. (We will deliver it only if we have time to finish it.)

Data

The data is generated by the simulation itself. Nicholas Molyneaux has been working on this since the last two years. The first set of data corresponds to the structures of the buildings. The second set of data corresponds to the simulated position, by time, of the pedestrians. The only "issue" is to be able to play with different datasets and stream

them through the visualization since the second set of data can be quite large (easily a few hundred of MBs). Thus, we decided to build a REST API to store the simulated data and stream them to the visualization.
The original data is composed of time, x and coordinates and pedestrian ID at irregular intervals. In order to make the visualization work in a smoother manner, we have interpolated the times in order to synchronize the times for all the pedestrians. This is the only pre-processing step which has been done. This is done on the back-end server, see section “Implementation & technical aspects”.

Main visualization (2D viz)

As explained earlier, the main visualization of this project consists in an interactive 2D visualization of the pedestrians, represented by dots, walking in a given structure. For the moment, the pedestrians have a fixed color but we will add different colors in function of different parameters. We will also add some overlaying maps. We started working on the density first. The density can be computed by using Voronoi cells using the pedestrians positions as the centroids. However, it is not possible to compute the Voronoi cells on the whole structure since the controlled area (in which the Voronoi cells are computed) has to be convex. Thus, we decided to let the user draw the controlled area.

Density

The sketch of the density layout is given below. We also have to use a specific colormap called the “Pedestrian level of service”, cited many times in the literature, as shown below..

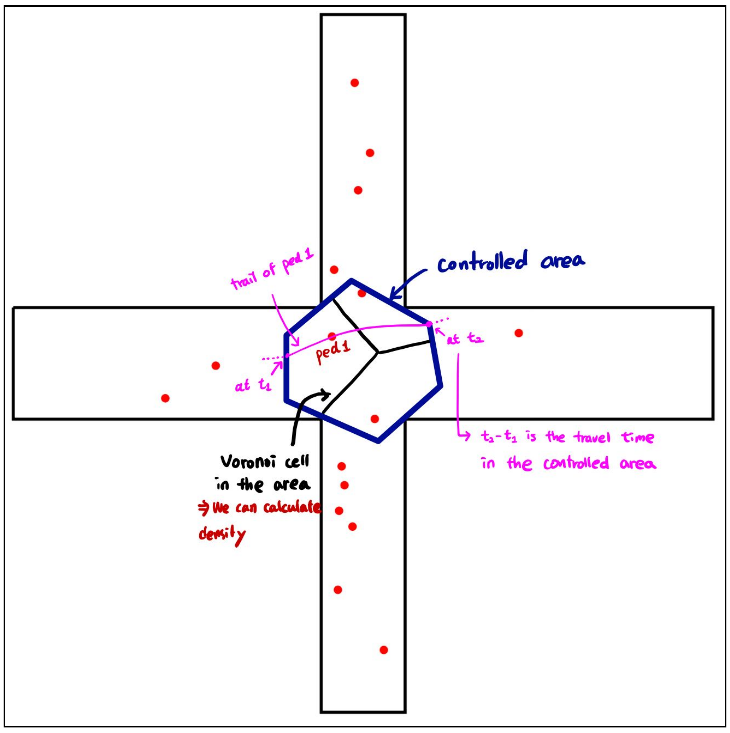
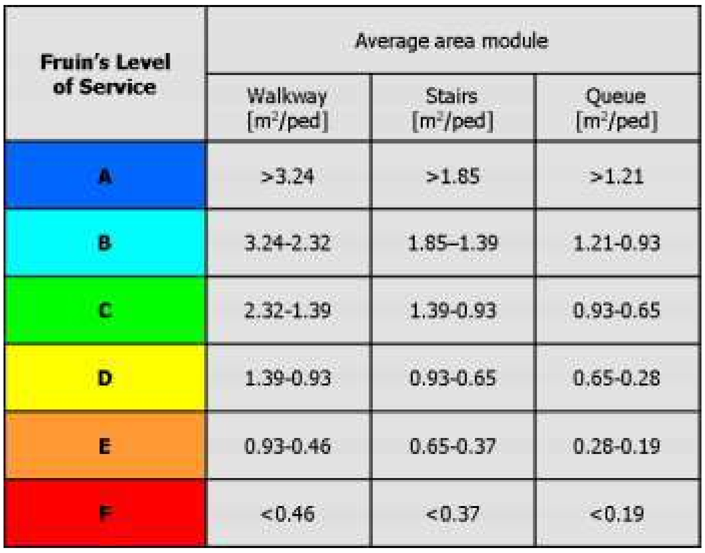

trail of ped 1
controlled area
ped 1
at t_1
at t_2
$t_2 - t_1$ is the travel time in the controlled area
Voronoi cell in the area
⇒ We can calculate density

Figure 2: Sketch of the density layout.



Fruin's Level of Service	Average area module		
	Walkway [m^2/ped]	Stairs [m^2/ped]	Queue [m^2/ped]
🟦 A	>3.24	>1.85	>1.21
🟦 B	3.24-2.32	1.85–1.39	1.21-0.93
🟩 C	2.32-1.39	1.39-0.93	0.93-0.65
🟨 D	1.39-0.93	0.93-0.65	0.65-0.28
🟧 E	0.93-0.46	0.65-0.37	0.28-0.19
🟥 F	<0.46	<0.37	<0.19

Figure 3: Pedestrian level of service.

The other layouts will be easier to draw and may be drawn on the whole structure or only in the controlled area.

Trajectories

The second overlay which we added is the trajectories of the pedestrians. This allows the user to visualize the areas of the infrastructure which are more heavily used then others. The goal is to achieve information like this:

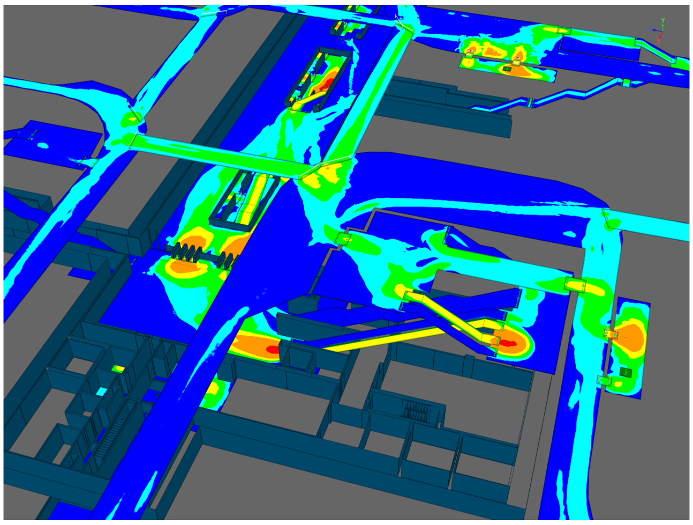


Figure 4: Heatmap of pedestrian usage, taken from https://architosh.com/2015/06/aia-show-floor-reports-oasys-massmotion-and-massmotion-flow/

In order to simplify the computation, we decided to overlay all the trajectories of the pedestrians. This way, by playing with the transparency, we can create a heatmap of the infrastructure. The final result of this overlay is this:

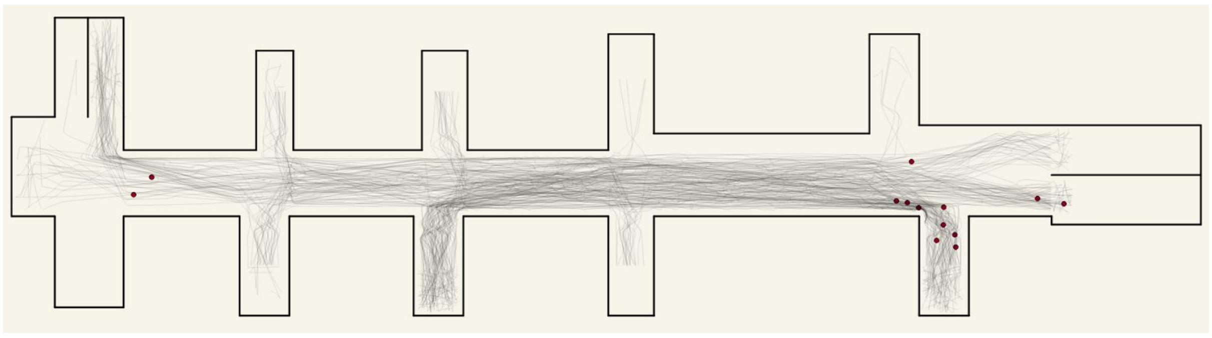


Figure 5: Trajectories visualization in the final website

Speed

The pedestrians can be coloured by their walking speed. This allows the user to see which pedestrians are moving slower or faster. The limitation of this approach is that since each pedestrian ha a different free flow walking speed, pedestrians will have different colours even though they are walking at their free flow speed. In order to improve this the data should include information about each pedestrians free flow walking speed. The final result is below:

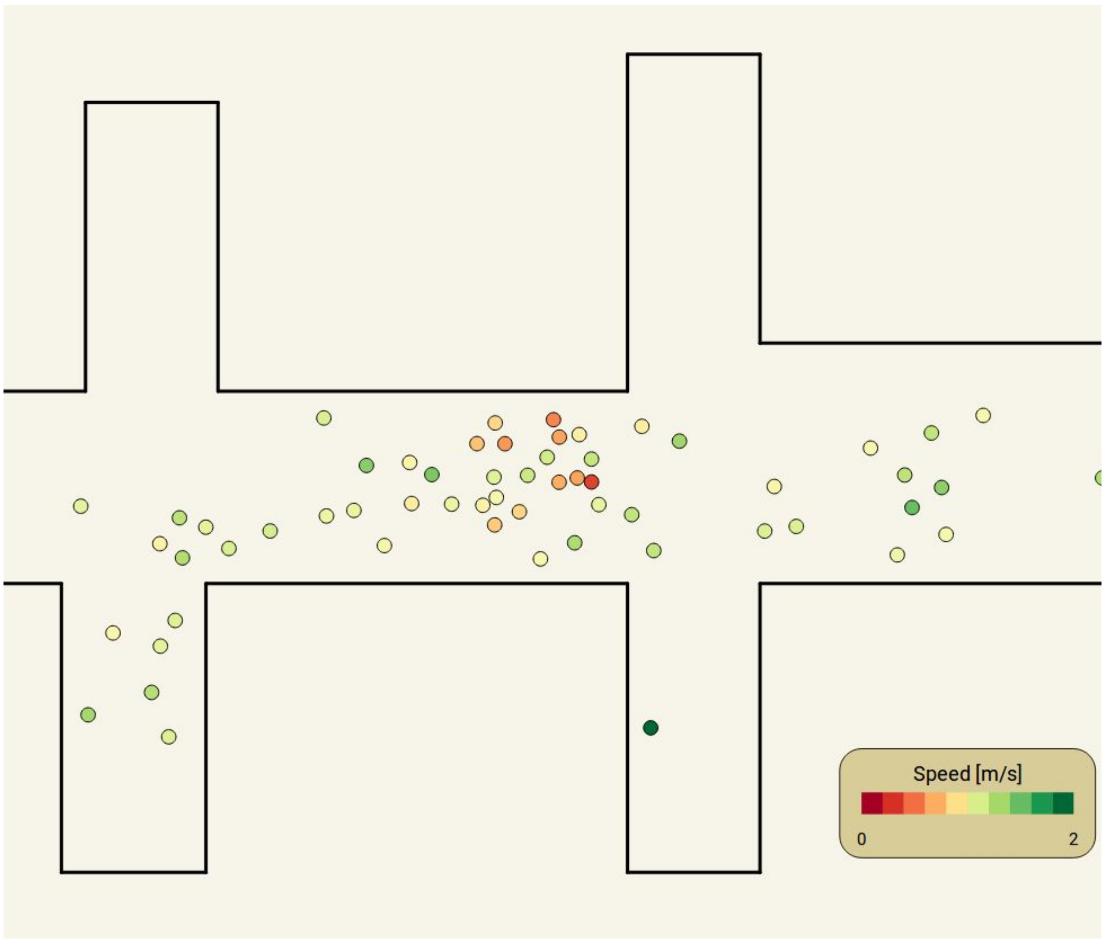


Figure 6: Pedestrians coloured by their walking speed.

Fun visualization (3D viz)

For this part, we want to create multiple, different pedestrians. Therefore, for the sake of simplicity, we decided that we will create Lego characters and animate them in Blender. Then, we will create different UV Mappings for the colors and randomly add them to the characters. The structure will be build in Three.js using the JSON files provided. (The same as for the 2D visualization.) Our intermediate state is given in Figure 9.

For the end of the project we have two different themes. The first is a “Walking data” theme with very scary pixelated zombies. The second is a more neutral theme where everybody is dressed the same. The idea was to present a story with the pedestrians being zombies in the morning. We attempted to overlay the densities and pedestrian trajectories in 3D but the result was bad. There were significant problems with the transparency of objects, hence we kept our 2D visualization for the more technical aspects.

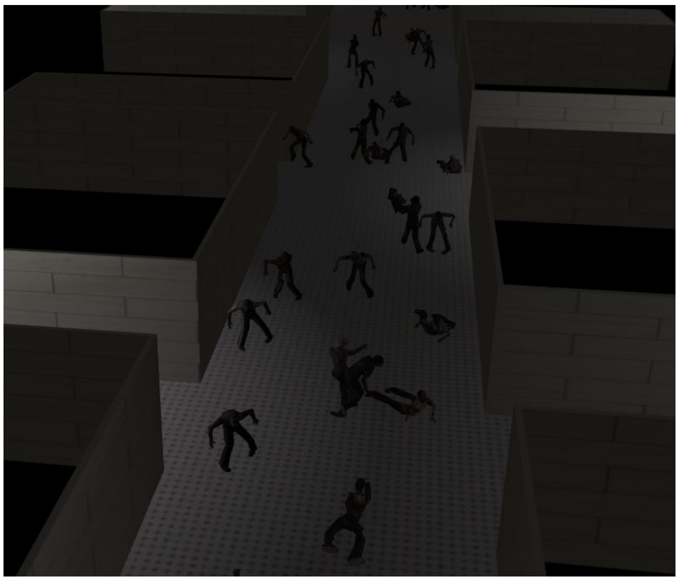


Figure 7: 3D visualization with the “Walking Data” theme.

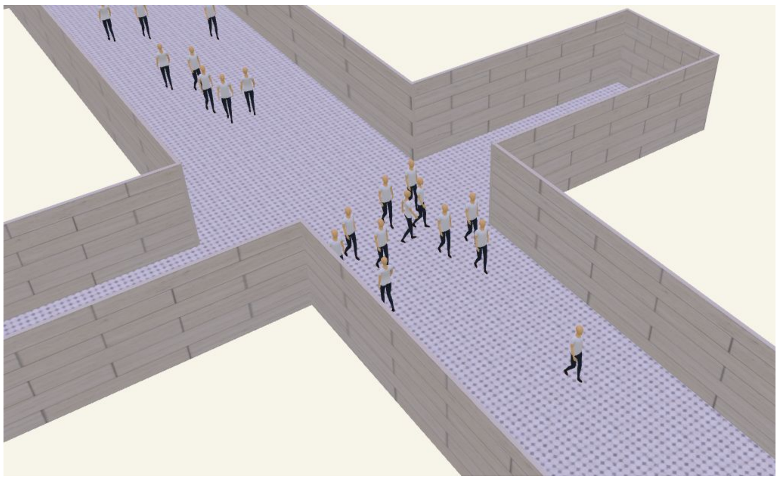


Figure 8: 3D visualization with the “normal” theme.

We also tried to combine the 3D visualization with the 2D visualization to give it more importance. We used a CSSRenderer (from THREE.js) to render the 2D as well as the WEBGLRenderer to render the 3D. However, the results, given in Figure 11, did not look good at all. Indeed, we were not able to hide the 2D visualization behind the 3D walls. In addition, the dots in the 2D visualization representing the pedestrians were blurry. So, we decided to keep the 3D visualization as is and use it to tell the story about the project and use the 2D visualization as the “real scientific visualization”.

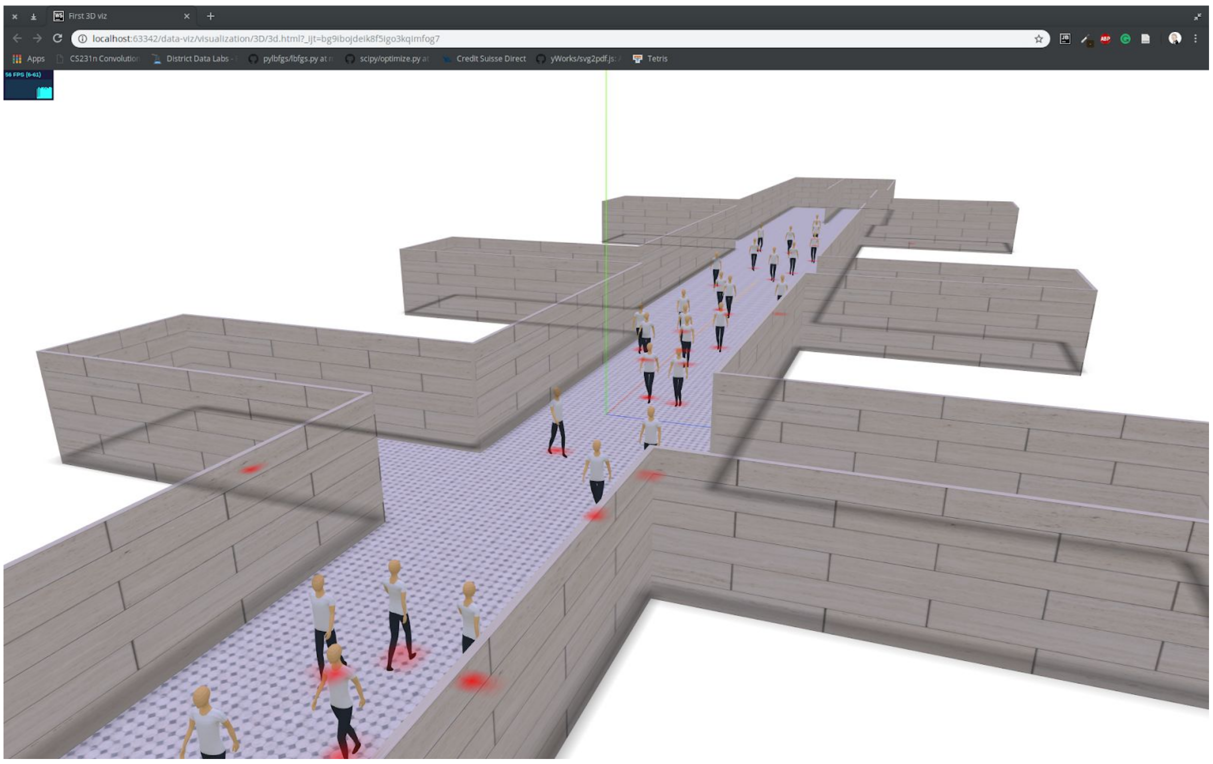


Figure 9: Failed attempt at combining the 2D and 3D visualization. The 2D dots are blurred and going through the walls. Thus, it does not look good.

Statistics

If we want to presents our results, we need to be able to get some static graphs such as the distribution of time travel, the distribution of speed, and the OD chord diagrams. These graphs are standard graphs that can be done with Matlab, Python, etc. Thus, we want to be able to modify the graphs even in Javascript. We, thus, decided to add the following options for the histograms:

- Change the color
- Change the number of bins
- Change the x and y axis values (min and max)
- Change the font size
- Export the histograms in SVG, PNG, and CSV.

The easiest way to add these options is to add an “option panel” next to the graph. For the OD chord diagram, we want to have a static graph. We initially planned on having a dynamic chord diagram representing the evolution of the origin and destination zones of the pedestrians inside the system. This was abandoned as it was hard to interpret correctly. Therefore the stats panel contains the OD chord diagram, travel time distribution and also density distribution.

The chord diagram representing the origin and destination of all the pedestrians inside the system looks like this:

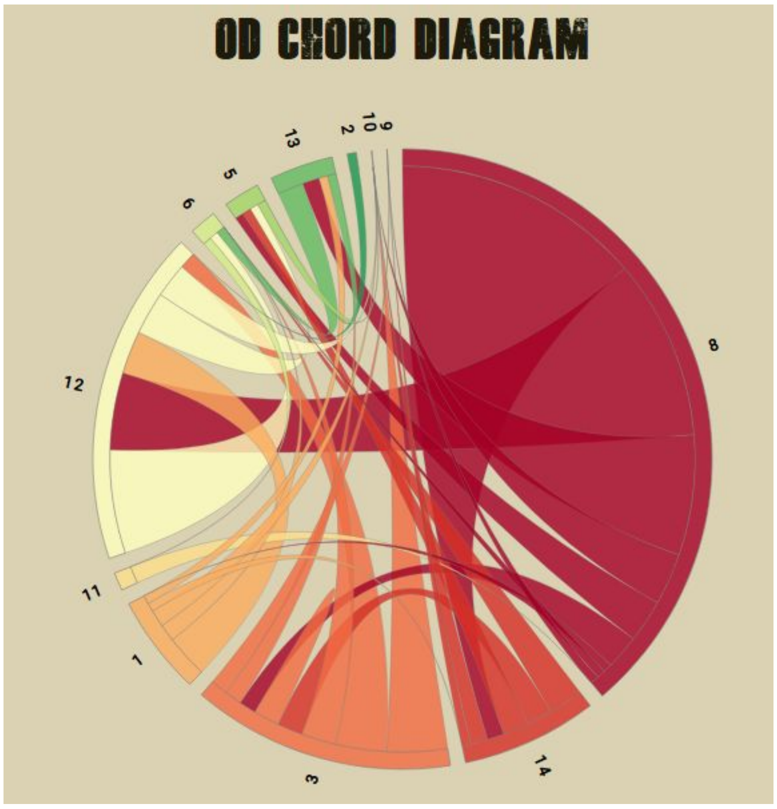


Figure 10: Chord diagram representing the origins and destinations of all pedestrians.

Website & Interactivity

Initial plan (i.e. not implemented)

The basic layout of the main page of the webpage is given on page 3. First, we want to show the title as well as some text to explain the visualization. We will also have a link to another page explaining the project in more details. We will also add the process book there. Then, we need to have two box talking directly with the API to get the available structures and data. Once the user has selected the data, we will show the main part of the visualization which is the 2D viz. We will have some options as well as a button to go fullscreen and a button to toggle the 3D viz. Below, we show the options for the statistics where the user can choose which statistics he wants to see. He will have to click on a button to fetch the data from the API. Then, for each graph, we will have some options available such as the color, the number of bins, the size of each axis, etc.

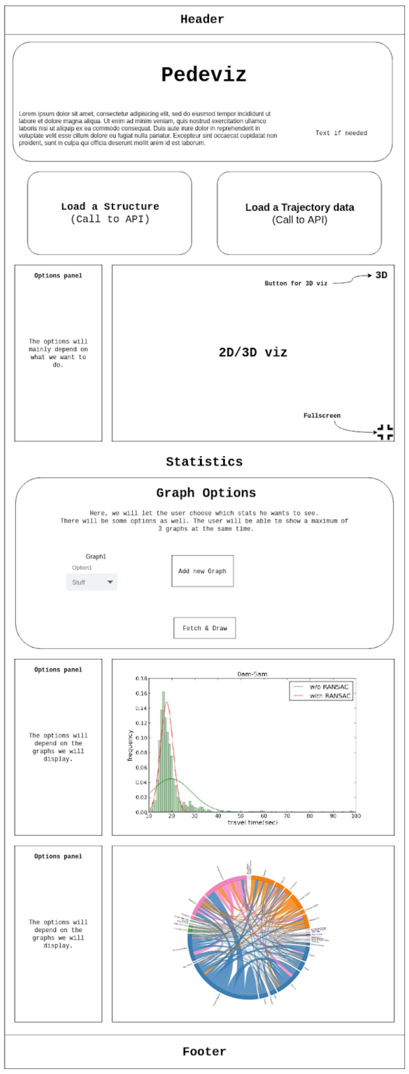
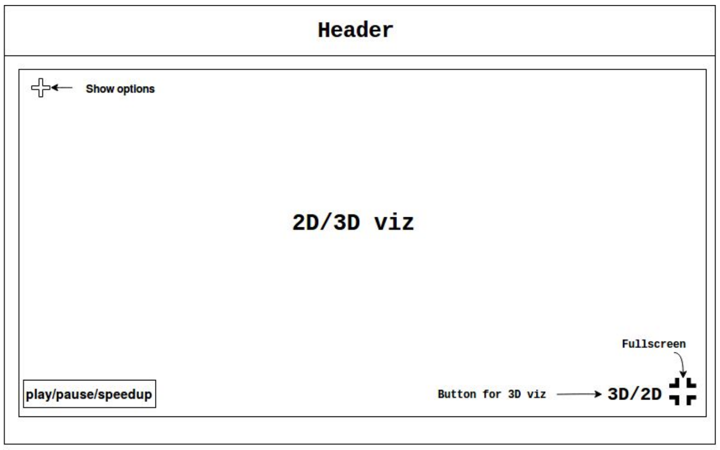

Header
Pedeviz
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
Text if needed
Load a Structure (Call to API)
Load a Trajectory data (Call to API)
Options panel
The options will mainly depend on what we want to do.
Button for 3D viz
3D
2D/3D viz
Fullscreen
Statistics
Graph Options
Here, we will let the user choose which stats he wants to see. There will be some options as well. The user will be able to show a maximum of 3 graphs at the same time.
Graph1
Option1
Stuff
Add new Graph
Fetch & Draw
Options panel
The options will depend on the graphs we will display.
0am-5am
w/o RANSAC
with RANSAC
frequency
travel time(sec)
Options panel
The options will depend on the graphs we will display.
Footer

Figure 11: Initial plan of the layout of the main page of the website.

In addition to this main page, we have an upload page. This page is used to upload new data to the API. We will also add the page explaining the project in more details and we'll provide some papers talking about the simulation. We may also add a page with the team members.

Final plan

As the project moved forward, we wanted to rationalize the interface so the users would find everything within the same screen and they would not need to scroll. Hence added a button to switch from the 2D to the 3D modes. All the options are hidden in a pane somewhere, the same the options for the stats charts (chord and histogram). The statistics became a pane which is hidden by default and can be shown by clicking on the dedicated button. When the window is resized, the statistics panel is moved below the main visualization.



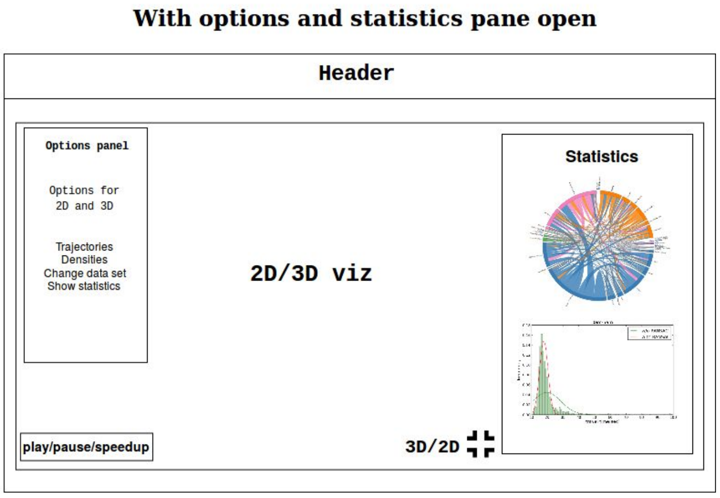


Figure 12: Final website structure. The top image is the default view where all options are closed. The bottom image is the position of the options and the statistics pane.

Interactivity

This section contains a list of all the possible interactions which are available for the different parts of the website.

Main visualization

- The main visualization in 2D and 3D require play/pause/speed up/slow down buttons as the objective is to be able to observe the movements of pedestrians inside the infrastructure over time. These buttons are located bottom left of the main visualization. The data used for the travel time histogram of the statistics panel is filtered by the time interval chosen in the “time bar”. The two extremities can be moved in order to select a subset of the data.
- For changing between the 2D and 3D views a button is also available. This has been challenging as the set of options is very different. The time must be coordinated between the 2D and 3D to avoid starting from the initial time. We

wanted to create a nice transition between both modes but this has proven very difficult and time consuming hence we abandoned the idea.
- The full screen button is located bottom right of the screen, next to the transition between 2D and 3D.

2D overlays

- The densities can be computed inside the control areas or inside a custom zone. The custom zone is drawn by clicking on the "Draw control area" button and then clicking in the viz to define the zone. A convex hull will be built from the points. This means a star cannot be drawn for example. This is a requirement for the computation of the voronoi diagram.
- The trajectories of the the pedestrian can be overlaid on the main 2D viz. Alternatively, the trajectories of specific pedestrians can be show by clicking on the pedestrian themselves (it is recommended to pause the simulation to make this process easier).
- Origin-destination filtering. The pedestrians with specific ODs can be highlighted by selecting the zones. Once the "zones" box is ticked, by cliking on a zone is becomes an origin, and by clicking on another zone while holding shift it becomes a destination zones. This filtering not only highlights the pedestrian in the main viz, but the travel time histogram is composed of the data from the filtered pedestrians.
- Speed. The pedestrians are coloured by their walking speed. The speed is computed using a forward finite difference scheme.

Change data

In the options when the 2D mode is selected, a possibility to change the data exists. This will load a different data to explore. As the new data will be sent over the network, one must wait for all the data to be loaded for all the options to be available.

Statistics options

These options are available by clicking the "plus" symbol next to the show statistics button.

- The chord diagram groups can be changed. By first clicking the "Group ODs" button, and the clicking on the name or the bar of the corresponding groups one wishes to aggregate together (the group is highlighted when the mouse is on the right place), the groups will be selected for "grouping". Then simply click on the Group ODs button again to rebuild the chord diagram with the new groups.
- The histograms have a set of options for changing their appearance. These are self explanatory and make allow fine tuning the graphs possible for exporting to use in slides !

Implementation & technical aspects

REST API & Database

The REST API is written in Scala with the “Play Framework”. The following sketch shows the how we want to integrate the REST API and the database with the visualization.

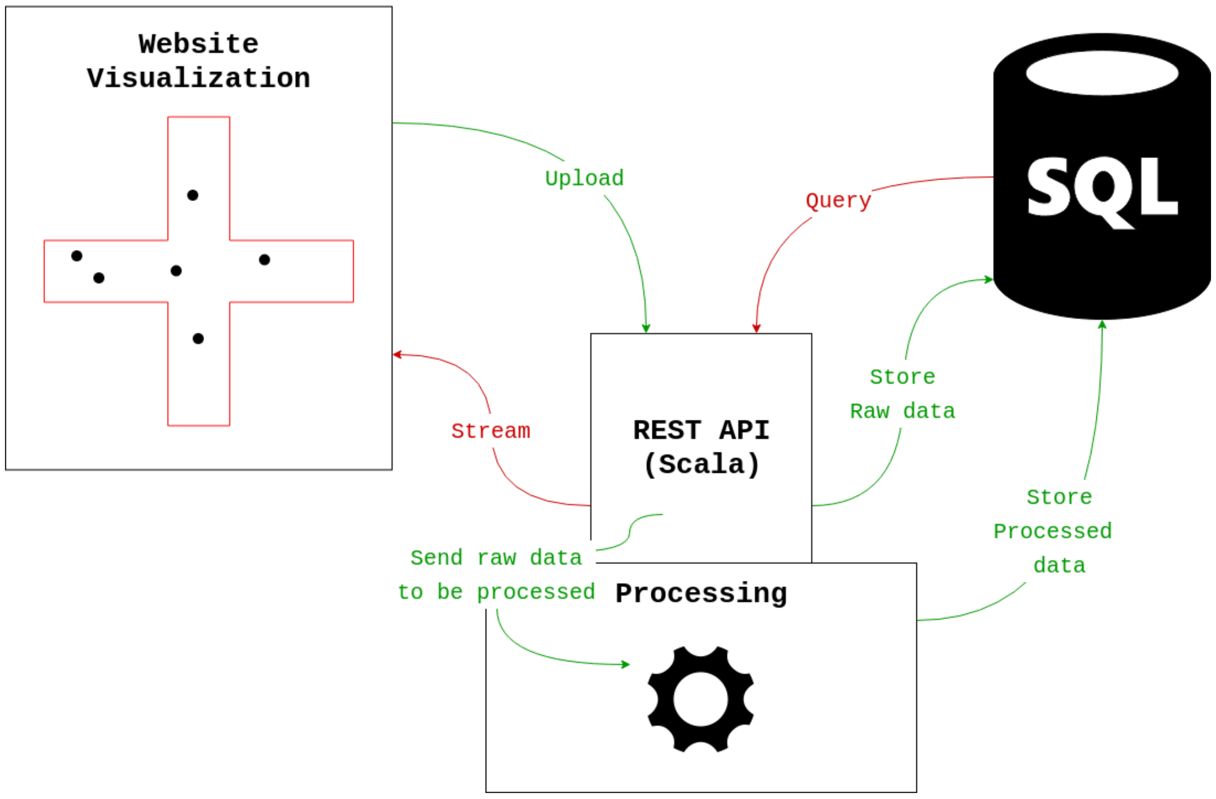


Figure 13: Flow of the visualization, the REST API, and the database.

As shown above, we need to be able to upload some new datasets from the website. Then, the API has to do some preprocessing before storing the data in an SQL database. Then, from the visualization, we need to be able to query/stream the data. The green arrows shows what we want to do when uploading a new dataset while the red arrows shows what is happening when we stream the data to the visualization.

Possible extensions

Some possible extensions to the visualization which could be added.

- Managing multiple floors. Assuming the information is available in the input data, each floor could be represented on top of each other in a nice way.
- Addition of elements used to control the pedestrian flows. As the simulations which create this data use control devices to manage the flows, representing these devices in the visualization would be interesting.

Known bugs

The visualization has mainly developed and debugged on Google Chrome. So, it is possible that some bugs remain, especially with other brothers.

For example, we know that the buttons on the visualization do not work with Touch Screens. The options is not draggable either. It is thus recommended to play with the visualization on a computer with the mouse and the keyboard.

Peer assessment

Nicholas Molyneaux mainly worked on the backend and as the team leader/coordinator. Indeed, since he will be the main user of this visualization, we always had to check with him if our ideas would be useful for him later on. He also worked on the Chord diagram and the down sampling of the trajectories. Sometimes, he can be stubborn (long discussion with Gael about some features), but we always found a middle ground.

Gael Lederrey started with the 3D visualization. Once it was working sort of fine, he went to work on the website structure using Bootstrap since he already had some experience with HTML, CSS, and bootstrap. He implemented the different communications between the website and the backend API as well as the histograms.. He can be really stubborn and sometimes have troubles delegating work. But everything was always done on time.

Semin Kwak worked on the 2D visualization. He implemented the whole 2D visualization with the different layers. He was always listening to Nicholas ideas and trying to improve them with his knowledge. He came up with many different ideas and always wrote his code such that it could easily communicate with the other parts of the visualization.

Conclusion: This project went very well. The team was able to work together without any big issues. Communication was easy and everybody was listening to the others ideas.

Contact

For any questions or major issues with the visualization, please send an email to nicholas.molyneaux@epfl.ch.