Deliverable 2.13

D2.13 An open, dynamic, high-resolution exposure model for Europe

Deliverable information	
Work package	[WP 2: Innovation: Exploiting innovation, technology advances and opportunities of big data for earthquake loss reduction]
Lead	[GFZ]
Authors	[Danijel Schorlemmer (GFZ), Fabrice Cotton (GFZ), Felix Delattre (GFZ), Tara Evaz Zadeh (GFZ), Marius Kriegerowski (GFZ), Lars Lingner (GFZ), Laurens Oostwegel (GFZ), Karsten Prehn (GFZ), Simantini Shinde (GFZ)]
Reviewers	[Ian Main]
Approval	[Management Board]
Status	[Draft]
Dissemination level	[Public]
Will the data supporting this document be made open access? (Y/N)	N/A
If No Open Access, provide reasons	
Delivery deadline	[28.02.2023]
Submission date	[28.02.2023]
Intranet path	[DOCUMENTS/DELIVERABLES/]

Summary

We created an open European, building-by-building and dynamic exposure model based on the engineering information from the *European Seismic Risk Model* (Crowley et al., 2020) and open data from OpenStreetMap. This European model is part of the Global Dynamic Exposure Model that aims at providing building-specific exposure data globally. This model combines engineering knowledge from existing classical exposure models with open data and predominantly OpenStreetMap to characterize every building as precisely as possible. It provides these exposure data in a fully open fashion, including the software system to generate it. To keep the model and its output open, only open input data is considered. The model is discretized to approx. 100m x 100m tiles of a global grid while still preserving the information for each building separately. The tiles are used to approximate building counts in case where OpenStreetMap data is incomplete. To preserve privacy, the publicly available data is aggregated to these tiles such that information about single buildings cannot be derived. However, the model input data and codes are openly available and users can create these data themselves. To create building-wise damage and loss assessments, we also provide a loss-calculator that aggregates losses by buildings or tiles.

The model is fully dynamic which means that it pulls new building data from OpenStreetMap every minute. These data are immediately processed and the exposure at the building locations is updated. Each building is represented by one or more assets describing in a probabilistic way what is known about this building. The model currently contains all European countries considered in the *European Seismic Risk Model*, most countries of South America and Japan. Within RISE, the European part of the model was planned to be finalized. More countries around the world will be included soon as open classical exposure models are available. The model data is available as country-excerpts in SpatiaLite databases for easy handling in QGIS or similar software. We also provide an API for damage and loss assessments based on the model data.

Introduction

With this deliverable, called the Global Dynamic Exposure (GDE) model, we aim to combine open engineering information about the global building stock with further detailed open data about buildings, mainly from OpenStreetMap (OSM), to increase the precision of exposure models. This increase is driven by the need to better model the spatial variations in ground shaking during earthquakes, to better understand each single building and the expected consequences for it from shaking, and to capture the constantly changing built environment, e.g. through urbanization and exchange of buildings after their end of life. Furthermore, pointing to the vulnerability modeling efforts within RISE, higher precision of building characterizations opens the path to building-specific vulnerabilities which cannot be realized with classical aggregated exposure models (AEMs). These models provide the exposure data aggregated over administrative regions; these data being the number of buildings of particular classes, their reconstruction value and number of people inside. These values are given separately for each building class per region. Thus they provide an overall picture of the expected building classes and their frequency and exposure indicators; however, their exact location, dimension, and further indicators adding to its specification are not part of such models.

The GDE model consists of six parts that we will describe in detail:

1. Generation of approx. 100m x 100m tiles and their properties related to the built environment as provided by the Global Human Settlement Layer (GHSL).
2. Building processing to understand all available properties from OSM.
3. Completeness estimation of the OSM buildings as compared to the GHSL built area per tile.

4. Spatial disaggregation of AEMs over the tiles.
5. Combination of spatially-distributed AEMs with building data.
6. The full dynamic chain to keep the tile, building, and exposure data up-to-date.
7. Export of static data excerpts.

In this deliverable, we prepare the European part of GDE. It is based on the *European Seismic Risk Model*[1] (ERSM20) as described by Crowley et al. (2020).

Tile Generation

Before we can distribute the AEMs over space, we need to understand the distribution of built area. Therefore, we imported the *Global Human Settlement Layer* (GHSL, Pesaresi & Politis, 2022; Schiavina et al., 2022), providing a first order distribution of built area for the entire globe. GHSL provides assessment of built area in approx. 10m x 10m cells as GeoTIFF. We vectorize this raster data into polygons and project them on a *Quadtree* level-18 grid. The *Quadtree* (Finkel & Bentley, 1974) is a common tiling system used by many Web Map Services (WMS) like Google Maps, Bing Maps, and OSM in combination with the spatial reference system EPGS:3857, the Web-Mercator projection (Battersby et al., 2014). The *Quadtree* sub-divides the square map of the world into four square-shaped tiles. At this first level of sub-division, called zoom level 1, the world consists of four square tiles. These can now be further sub-divided, increasing the number of tiles each time by a factor of four. Here, we use level-18 tiles which are approximately 100m x 100m in size in Europe. Globally we have more than 68 billion tiles of this zoom level, see Fig. 1. We store each tile with its associated GHSL polygon in our tile database[2], see Fig. 2. These tiles have been chosen to provide fast and easy access to the exposure data but also to preserve basic privacy when distributing data.

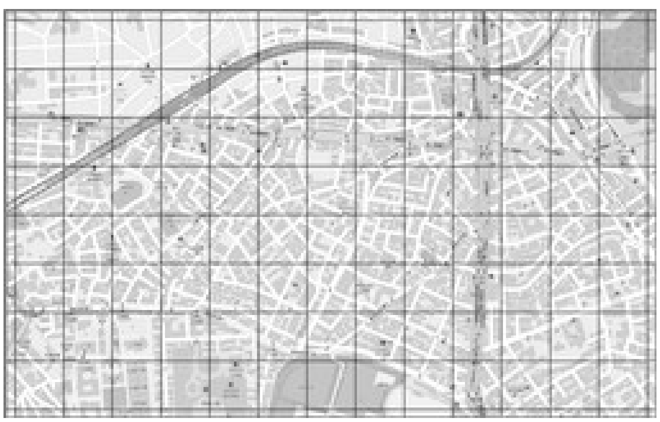

Figure 1: Tiles in the area of the Yoyogi-Uehara train station in the Shibuya district of Tokyo, Japan. Background map: copyright OpenStreetMap contributors.

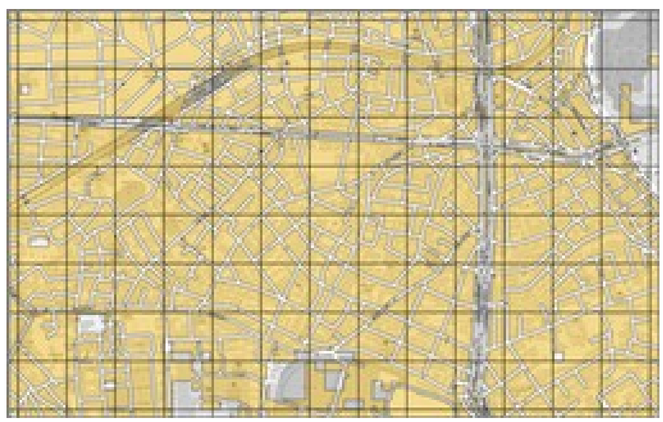

Figure 2: Map as in Fig. 1 with the built-area polygons from GHSL (with roads cut out). Background map: copyright OpenStreetMap contributors.

In the next preparatory step, we obtain all necessary boundaries of all administrative regions for which AEMs provide exposure data. The ESRM20 uses regions defined by administrative names matching the boundary data from GADM[3]. We found many problems in the GADM dataset, mainly low resolution boundary polygons, non-matching polygons between different administrative levels or along country borders, and restrictions in distributing this dataset; the latter rendering them to be not fully open. To develop our model on the best possible

1 https://gitlab.seismo.ethz.ch/efehr/esrm20/-/archive/main/esrm20-main.zip
2 https://git.gfz-potsdam.de/dynamicexposure/openbuildingmap/database-obmtiles
3 https://www.gadm.org

and open boundary dataset, we extract high-resolution boundary polygons from OSM using our tool *Borderline*[4] and store them in our tile database.

Building Processing

We operate a copy of OSM data that is kept up-to-date on the basis of the so-called *minutely diffs*. These files, released every minute, describe the changes to the overall OSM database. With every change, our tool *Spearhead*[5] checks which buildings have been added, changed or deleted, and submits them to our rule-based processing engine *Rabotnik*[6]. This engine processes all incoming buildings asynchronously and distributes the load to as many processors as are assigned to it. During this process, we determine for each building separately several exposure indicators such as occupancy and floor space based on the implemented rules. The most important rule assesses the occupancy on the basis of the underlying land use of a building, the building tags and points of interest within the building which can point to specific uses of the building. Further rules extract the number of stories or calculate this number from a given building height. With the number of stories known and the building footprint geometry given, we compute the floor space of the building. Further exposure indicators indicating the roof shape, the overall shape of the building, and the number of adjacent buildings will be implemented soon as new rule sets. These indicators are then used to create the taxonomy description as defined in the GEM Building Taxonomy[7] to describe the type of the building.

The building database currently contains around 550 million buildings and is growing by approx. 5 million buildings a month or 2 per second. The operational system can process several 100 buildings per second so that it can easily keep up with the growth of data.

Completeness Estimation of Tiles

We have to account in our model for the fact that although OSM is the largest geographic dataset, its collection of building data is not complete. While several countries already have the complete building stock in OSM (partly through imports of open cadastral data, such as France, the Netherlands, etc.), other countries may lack significant portions of these data. With a database containing all tiles and the size of the built area as specified by GHSL and all buildings from OSM, we can estimate the completeness level of each tile. For this, we have conducted a manual completeness analysis for the area of Attica, Greece by visually comparing the building data in OSM with satellite imagery. Each tile was classified as either complete or incomplete if buildings were present in the satellite imagery, otherwise as empty or undecidable in case of cloud cover. We have developed an empirical function comparing the size of the built-area polygon in the tiles taken from the GHSL with the total size of the building polygons taken from OSM in the same tile. If a threshold in the ratio is exceeded, the tile is considered complete. Our tool *all-in*[8] processes all tiles and provides the respective completeness assessment. In case a tile is considered complete, the built area used later for processing is set to the sum of the area of the building-footprint polygons in the tile.

4 https://git.gfz-potsdam.de/dynamicexposure/borderline
5 https://git.gfz-potsdam.de/dynamicexposure/openbuildingmap/spearhead
6 https://git.gfz-potsdam.de/dynamicexposure/rabotnik/rabotnik
7 https://gitlab.seismo.ethz.ch/efehr/esrm20/-/archive/main/esrm20-main.zip
8 https://git.gfz-potsdam.de/dynamicexposure/openbuildingmap/all-in

Figure 3: Map as in Fig. 2 with building footprints from OpenStreetMap. In the upper-left corner one can clearly dense built area without building footprints, indicating missing building data. Background map: copyright OpenStreetMap contributors.

Figure 4: Map as in Fig. 2. The tile built area is colored by building completeness. Green and red indicate complete and incomplete tiles, respectively. Background map: copyright OpenStreetMap contributors.

In some countries in Europe, the OSM community has either fully imported cadastral data or mapped all buildings. The building data for France and the Netherlands have been fully imported and thus we have assigned all tiles in these countries manually to complete. This results in each tile now representing the size of building-footprint polygons as the built area and the built-area estimate from GHSL is ignored. Besides these two countries, we identified Austria, Czechia, Denmark, Estonia, Germany, Iceland, Luxembourg, Liechtenstein, Monaco, Poland, Slovakia, and Switzerland as also being complete. While these countries may still lack a handful of buildings, the overall error in exposure data is strongly reduced by only using built-area assessments based on the buildings in the database over GHSL.

Spatial Distribution of AEM

Before we can combine any of the AEMs with building data, we disaggregate them spatially. The AEMs store a set of assets per administrative region, where each asset describes one building class, its frequency, total reconstruction value and number of people inside. We select for each region all tiles within their respective boundary, compute the total built area (either based on GHSL or the building footprints depending on the completeness level), and assign each asset to all tiles proportionally to the tile's built area, creating so-called tile entities for each tile. At this stage, all assets are homogeneously distributed and each tile contains the same assets with the same proportions. Thus, this represents a simple spatial disaggregation based on the built-area per tile. This process is performed with our tool *Initializer*[9] and the result we call the *baseline model*.

Combination with Building Data

The key process in the entire GDE chain is the combination of the spatially disaggregated AEM with the building data enriched with exposure indicators. For this task, we iterate through all tiles with AEM data in it. In the first step, we take the assets as imported from the AEM and consider it the final exposure data for this tile, see Fig. 5. Then we iterate through all buildings and assign them the combinations of assets as defined in the baseline model of the tile. Hereby we check each building-class taxonomy of the baseline model for compatibility with the exposure indicators of the building and only keep the ones that match the exposure indicators. This usually leads to a reduced of set of assets that probabilistically describe the possible classes for each building, see Fig. 6. We normalize the remaining assets per building to represent exactly one building and store this in building entities. Simultaneously, we reduce the remaining assets in the tile entity by proportionally subtracting one building. In general, this final model consists of one tile entity and zero to several building

9 https://git.gfz-potsdam.de/dynamicexposure/globaldynamicexposure/exposure-initializer

entities per tile, see Fig. 7. The tile entity possibly contains assets that have not been encountered for by the buildings, e.g. a tile being assigned 7.4 buildings according its GHSL built area but only 6 buildings were present in the database. The remaining 1.4 buildings and their assets are kept being assigned to the tile entity. If the tile is considered to be complete, the remaining 1.4 buildings are deleted. Also, should the number of remaining buildings be less than zero (more buildings found than expected), this number is set to zero. This process of combining buildings with the AEM results in reducing the number of possible building classes per building from the initial number representing the classes per region as defined in the AEM to the ones that match the building properties. This step is run with our tool *Finalizer*[10].

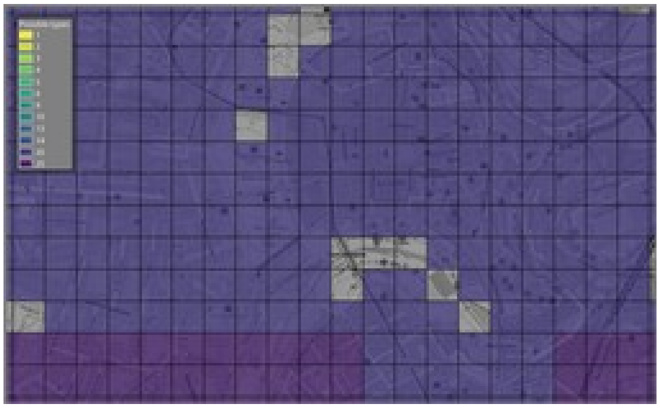

Figure 5: Number of possible building classes per tile according to the ESRM20 model in Luxembourg. Background map: copyright OpenStreetMap contributors.

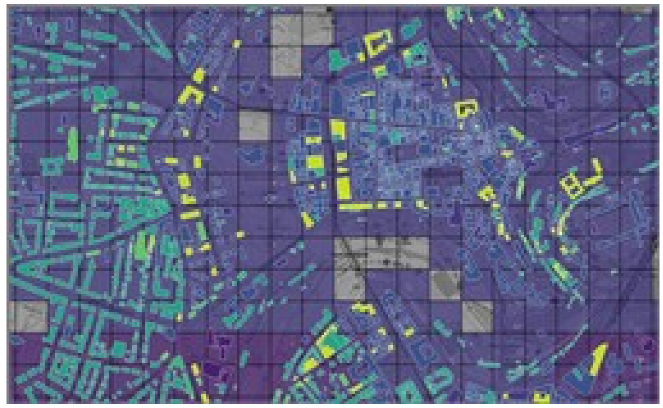

Figure 6: Maps as in Fig. 5. Additionally the buildings with their number of possible classes are plotted on top. The reduction is clearly visible. Background map: copyright OpenStreetMap contributors.

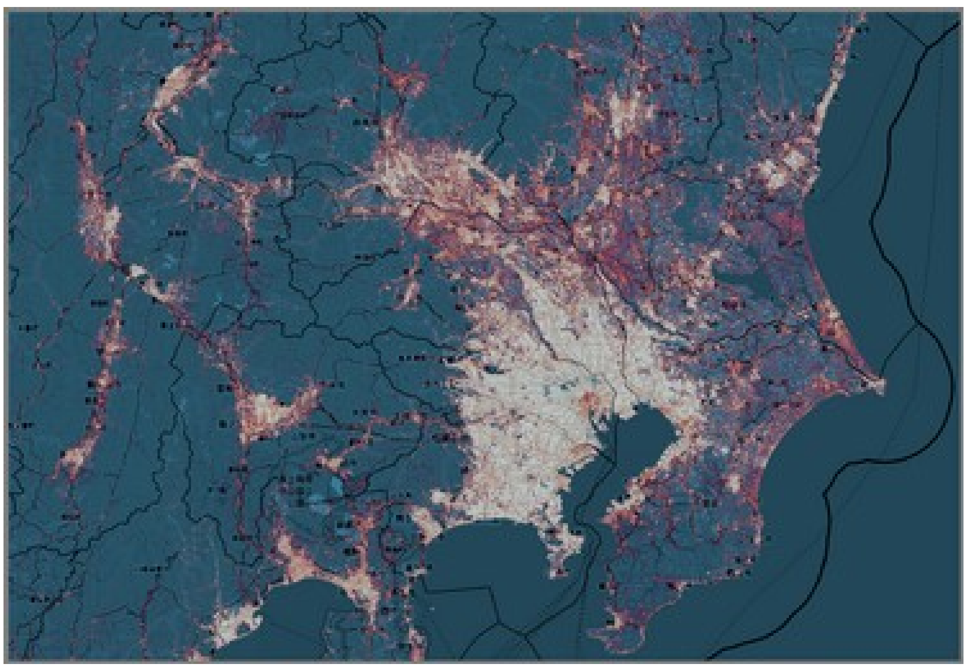

Figure 7: Exposure model for Japan. The number of buildings per tile in the Kanto region is displayed in color. The lighter the color, the more buildings are located in the tile. Background map: copyright OpenStreetMap contributors.

10 https://git.gfz-potsdam.de/dynamicexposure/globaldynamicexposure/exposure-finalizer

Full Dynamic Processing Chain

As mentioned above, OSM is changing constantly with approx. 2 buildings being added per second in average, while others are modified. To keep up with these changes, we developed our processing engine *Rabotnik*. We already described how *Rabotnik* keeps the building database up-to-date by processing every new, changed, or deleted building. As soon as a building has been processed and inserted into the database, its corresponding tile is identified and the tile ID is submit to another *Rabotnik* processing queue. This one estimates the completeness of the tile that may have changed based on the change of this one building. It applies the same rule set as the *all-in* tool. With this part of the dynamic chain, we keep our tile database up-to-date and have the latest completeness at hand for each tile globally.

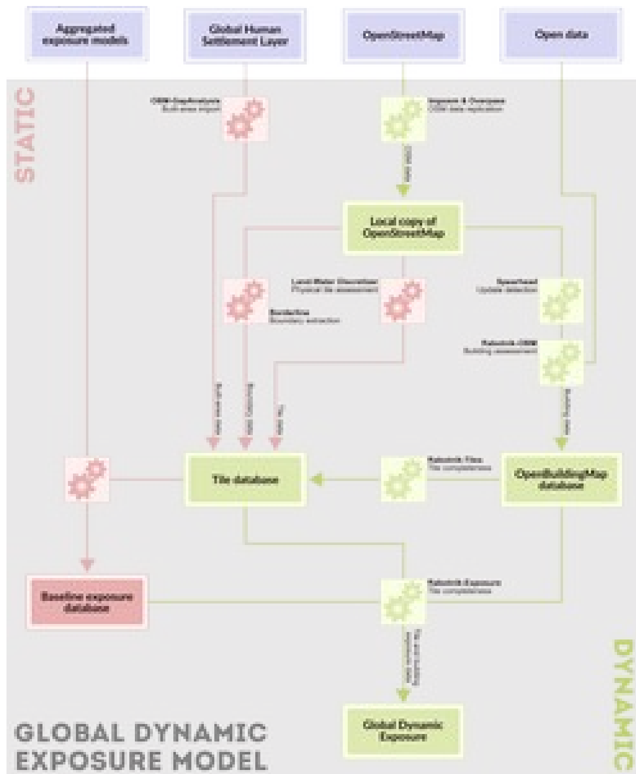

Figure 8: Schema of the Global Dynamic Exposure model. Static elements and processes (run once) are displayed in red, dynamic ones running with every change in OpenStreetMap are shown in green.

The final dynamic step again employs our processing engine. After the completeness of a tile has been computed, the tile ID is passed on to another *Rabotnik* queue that resembles the process of *Finalizer* but for one tile only. This process combines the static baseline model from the AEM per tile with the most recent building data and the new completeness estimate computed seconds before.

If somebody adds or modifies a building in OSM and uploads their changes, it takes about five minutes for OSM to provide the respective minutely-diff file containing the uploaded change. As soon this file is released our replication process downloads it and the computational chain is triggered, see Fig. 8. It takes less than 2 minutes until the exposure values in the respective tile of the changes building have been updated, making our model highly dynamic. This short turn-around times are helpful in case of disaster-crisis management as any crowd-sourced building mapping continuously updates the exposure model and helps refining the damage and loss assessments.

Export of static data excerpts

Regarding the dynamic nature of GDE and to facilitate the needs of researchers for reproducible models that can be referenced, we create special data excerpts. For RISE, we prepared a data excerpt reflecting all input data as they were on 1 January 2023, 00:00 UTC. For this, we have created an OSM extract of exactly this time and run the entire processing chain with this data. This results in a European exposure model reflecting the building data of 1 January 2023. This way, others can rerun our codes with this clearly specified dataset to reproduce our results.

Furthermore, from this model we are creating country extracts that are available on Zenodo via a DOI. Given the privacy-relevant data we produce, the downloadable excepts do not contain any specific building information anymore. Instead we aggregate the assets of all building and their tile entity into the tile entity to mask the single buildings and their geometry. If tiles contain too few buildings, we aggregate them further into large tiles to make sure no conclusions about particular buildings can be derived. The result is in general a multi-resolution

Quadtree tile grid, with its resolution depending on the building density, see Fig. 9. This tile aggregation process, following the *Quadtree* approach by decreasing the zoom level, can also be directly applied to reduce the resolution for fast loss assessments. All excerpt processes can be run with our tool *Share*[11].

Damage and Loss assessments

To enable the user of our model to run damage and loss assessments with the model, we developed the *Loss-Calculator*[12] and additionally an API that can be used for smaller-scale assessments. The API takes the basic parameters of an earthquake and uses the *ShakeMAPI*[13] to compute a ground-motion field for the earthquake. Depending on the extent of the ground-motion field, a suitable exposure model is retrieved from the main database and stored as data excerpt in a SpatiaLite database. The user can choose the level of aggregation for this excerpt. The more tiles are aggregated into parent tiles, the faster the loss assessment will be computed; likewise the full resolution is possible and all buildings will be used separately for any damage or loss assessment. The Loss-Calculator computes the estimated damage probabilities and losses using the fragility and vulnerability functions that the user provides as either CSV or XML files (discrete or continuous fragility functions). The results are aggregated to each building and tile so that various visualizations are possible. The damage and loss values are also stored in the same SpatiaLite database that contains the exposure data. This database is augmented with special database views about various results and data properties that makes it easy for the user to visualize them in QGIS, see Fig. 10.

11 https://git.gfz-potsdam.de/dynamicexposure/globaldynamicexposure/exposure-share
12 https://git.gfz-potsdam.de/dynamicexposure/globaldynamicexposure/loss-calculator
13 https://git.gfz-potsdam.de/loki/shakemapi

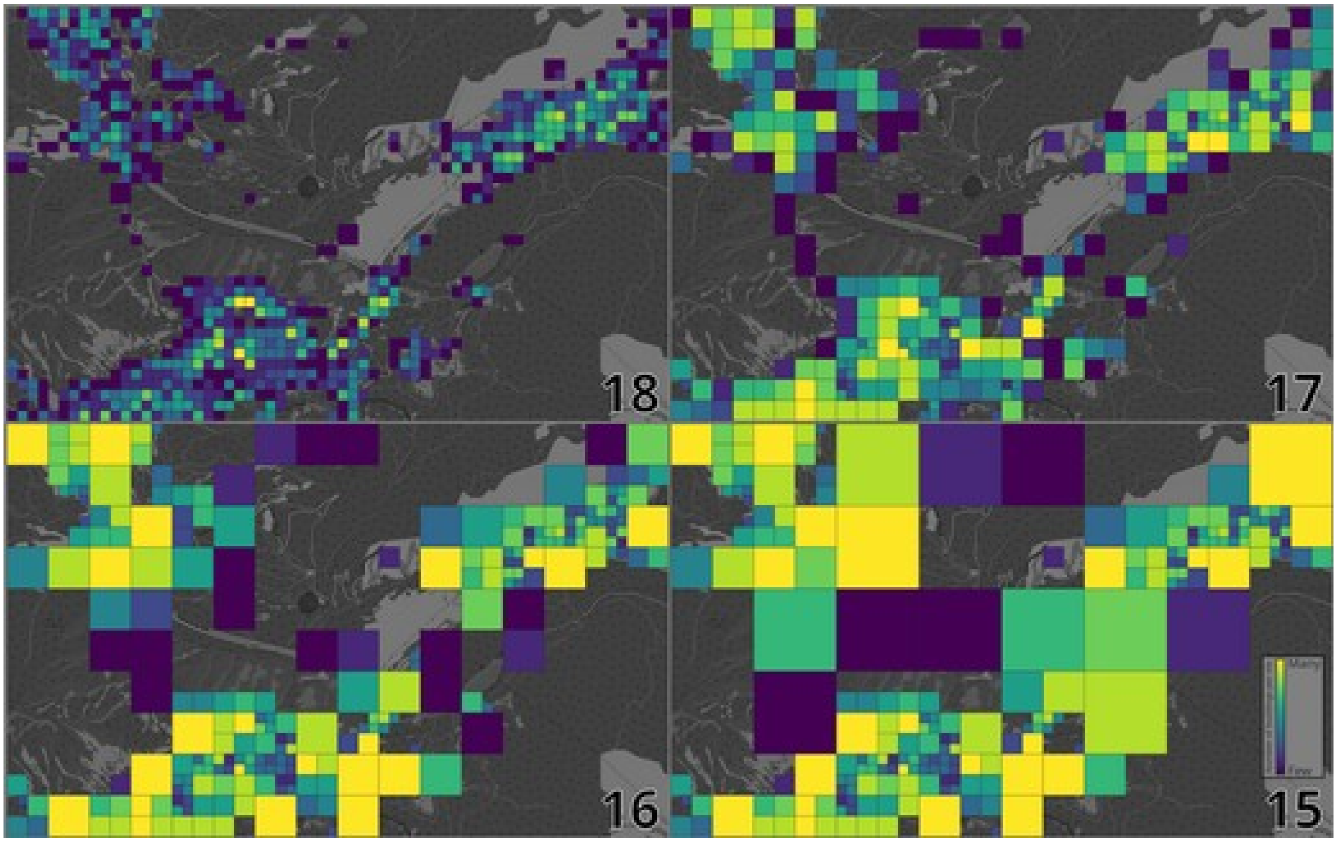

Figure 9: Different levels of aggregation of the exposure model in Andorra for protecting privacy. Colors indicate the number of buildings. The purple tiles contain fewer buildings than a defined threshold and are joined with their neighboring tiles to parent tiles of lower zoom level. The top left frame shows the initial model data aggregated on zoom-level 18 tiles. The other frames show the model aggregated down to the minimum zoom level as indicated in the bottom right of each frame. The bottom-right frame shows how the number of tiles with low number of buildings has decreased while the high resolution of tiles was preserved in areas with many buildings. Background map: copyright OpenStreetMap contributors.

Technology

The entire software chain is implemented in Python and fully open source. All codes are released under the AGPL. For data storage, we use PostGIS database running in Docker environments. The data structure of these databases is fully documented in database migration scripts. All codes, scripts and docker definitions are published on our GitLab[14]. For local data excerpts, we use SpatiaLite databases. For visualizations, we use QGIS and have developed plugins for easier handling of our data. All codes we developed are open source and licensed under the AGPL. All software we rely on is also open source. No proprietary software is needed to implement and run the model.

14 https://git.gfz-potsdam.de/dynamicexposure

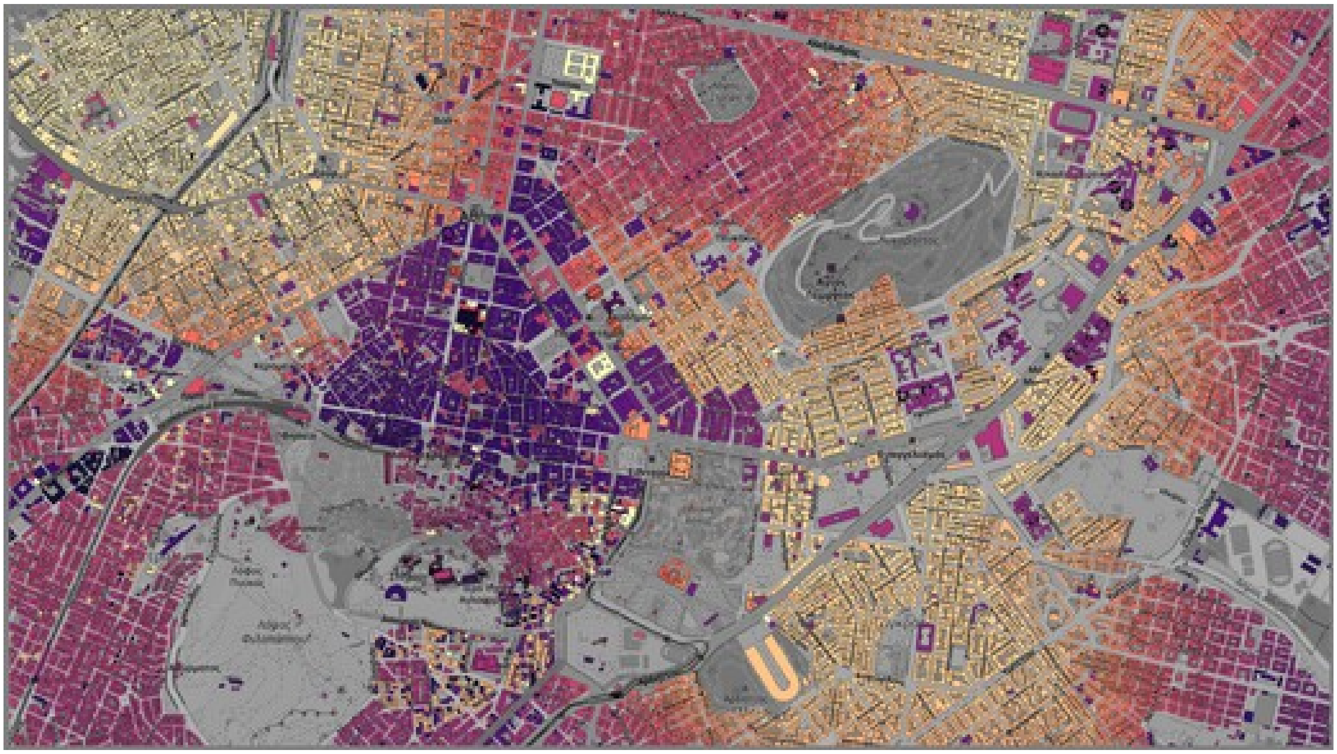

Figure 10: Damage assessment in the city of Athens for an earthquake northwest of the city. Buildings are colored by their probablity for experiencing slight damage. The brighter the color, the higher the probability. Both, the spatially varying ground-motion levels and the specific building characteristics influence the expected damage grade probabilities. Background map: copyright OpenStreetMap contributors.

References

Battersby, S. E., M. P. Finn, E. L. Usery, and K. H. Yamamoto (2014), Implications of Web Mercator and Its Use in Online Mapping, Cartographica 49:2, 2014, pp. 85–101, 10.3138/carto.49.2.2313

Crowley H, Despotaki V, Rodrigues D, et al. Exposure model for European seismic risk assessment. Earthquake Spectra. 2020;36(1_suppl):252-273, 10.1177/8755293020919429

Finkel, Raphael & Bentley, Jon. (1974). Quad Trees: A Data Structure for Retrieval on Composite Keys. Acta Inf, 4, 1-9, 10.1007/BF00288933.

Pesaresi M., Politis P. (2022): GHS built-up surface grid, derived from Sentinel2 composite and Landsat, multitemporal (1975-2030), European Commission, Joint Research Centre (JRC)
PID: http://data.europa.eu/89h/d07d81b4-7680-4d28-b896-583745c27085, 10.2905/D07D81B4-7680-4D28-B896-583745C27085

Schiavina M., Melchiorri M., Pesaresi M., Politis P., Freire S., Maffenini L., Florio P., Ehrlich D., Goch K., Tommasi P., Kemper T. (2022). GHSL Data Package 2022, JRC 129516, ISBN 978-92-76-53071-8, 10.2760/19817