# Interactive Design of Noise Protection Walls

Fabian Pfitzner[1], Ivan Bratoev[2], and Štefan Jaud[1]

[1]Chair for Computational Modeling and Simulation · Technische Universität München · Arcisstraße 21 · 80333 München · E-Mail: fabian.pfitzner@tum.de (Pfitzner); ga24yuk@mytum.de (Jaud)
[2]Chair of Architectural Informatics · Technische Universität München · Arcisstraße 21 · 80333 München · E-Mail: ivan.bratoev@ai.ar.tum.de

The design of noise protection walls alongside an infrastructure asset currently involves long hours of simulation and complex computations. This represents a huge burden in the early design stages, where all information is vague and an emphasis is laid on quick comparisons rather than precise results. This paper presents an interactive noise simulation tool to be used in the early design stages. It expands the Collaborative Design Platform (CDP) and builds upon the Interactive Alignment Plugin presented at the $30^{th}$ Forum Bauinformatik. The main aim is to enable intuitive and quick evaluation of design variants of positions and dimensions of noise barrier walls. The noise calculation is performed according to the RLS-90 norm which provides simplified formulas for quick estimation of traffic noise on roads. The results of the developed simulation tool were tested with examples provided in the RLS-90 norm and show agreement within 1 % deviation.

**Keywords**: noise simulation, BIM, infrastructure

## 1 Introduction

### 1.1 Motivation

Conventional methods of noise calculation often involve complex, long-term calculations. On the one hand, this is due to the large number of the influencing parameters and, on the other hand, the requirements that are too high. Especially in early design stages, it is often unnecessary to include every small detail in calculations. It is rather important to create a quick overview in order to recognise whether a construction measure is necessary or not. For carrying out fast noise calculations alongside the road, only a few parameters are required. In order to avoid irrelevant factors, a number of the characteristics are neglected, e.g. the surface quality of the road.

In the area of noise simulation there are many points for involving optimisation concepts. The motivation for developing this plugin resulted from improving old obsolete proceedings, described above. A simple and fast noise calculation is a great relief for early design decisions and could fundamentally change the planning of structural noise barrier systems in the future. Furthermore the process of incorporating Building Information Modelling (BIM) methods in the construction industry can be enhanced.

### 1.2 Collaborative Design Platform // CDP

The CDP provides the ability to perform real-time computer analysis intuitively. The table is based on two principles. On the one hand, it promotes the embedding of the creative process. This allows simulations and analyses to be carried out even in the early design stages. On the other hand, it also provides easy creation and design of user-friendly tools. An example of this is the visual programming language, which enables the user to perform analyses and simulations without implementing code Schubert and Petzold (2017).

The basic idea of the table presented in Figure 1 is to provide an interactive platform for early work phases. The focus lies on physical models, such as blocks, with a touch surface. Changes in the model have a direct influence on the simulation in real time and the calculated results are displayed instantly (Schubert, 2012).

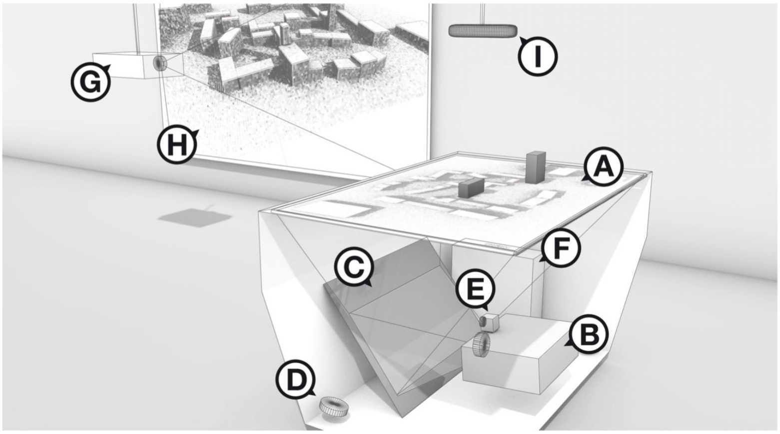



**Figure 1:** CDP-Table with matt projection surface (A), projection image (B), mirror (C), infrared emitters (D), infrared camera (E), computer (F), projector (G), and screen (H) (Schubert, 2012).

The hardware was developed with the above principles in mind. The system is self-made and offers a multi-touch surface as a basis. With the help of the Diffused Illumination (DI) technology the recognition of any objects is made possible. The interactive table has a matt projection surface on which an image is projected from below and redirected via a mirror. The projection surface is illuminated with infrared emitters. The underside of the projection surface is recorded via the mirror with an infrared camera. Objects and touches are visible on the infrared camera image. Using the Microsoft Kinect camera, objects placed on the table can be scanned as a 3D point cloud. In combination with the camera image it is possible to create a 3D model (Schubert, 2012).

Finally, a computer processes the camera data for the projection image. In addition to the display of two-dimensional information, it is also possible to display a three-dimensional representation of the design for better understanding. The plugins are connected via middleware (written in C++). Data communication, interaction as well as visualisation of the calculation results run completely through the host application, which provides the required data (Schubert, 2012).

## 2 Basics of Noise Calculation

Noise can be described with elasto-dynamic waves and vibrations. A human being can perceive noise as sound pressure in a range of 0.0002 to 20 Pascal (NALS/NATG, 2009). The sound pressure can be logarithmically adapted to the sound pressure level. In the RLS-90 standard, the sound level is characterised as sound pressure level and defined in decibel [dB] (Bundesministerium für Verkehr, 1990). Basically, all noise levels are evaluated depended on frequency, due to the sensitivity of the human hearing (Häupl et al., 2017).

Noise calculations according to the road centreline are carried out using average noise levels, differentiated between night and day cycles. It is also important to consider terms like emission (the emitting noise location) and immission (the exposure noise location) (Bundesministerium für Verkehr, 1990). The different noise levels can be illustrated with isophones which are curves with the same noise level and are derived from the 40-phon isophone (Häupl et al., 2017).

For receiving proper results of noise calculations a special case of the conventional methods RLS-90 norm has to be applied. Therefore the slicing method is used, which divides the curve into many sections (Bundesministerium für Verkehr, 1990). The number of sections depends on the smallest distance $s$ between the location of immission and the location of emission. It must be pointed out that a maximum section length of 0.5 s travel time is allowed. The medium level $L_{m,i}$ for a single section $i$ can be described with:

$$L_{m,i} = L_{m,E} + D_l + D_s + D_{BM} + D_B \ , \tag{1}$$

where $L_{m,i}$ stands for average level of segment $i$, $L_{m,E}$ for emission level, $D_l$ for adjustment considering the section length, $D_s$ for distance and air absorption, $D_{BM}$ for soil and meteorological attenuation, and $D_B$ for topographical and structural conditions.

The components of (1) have to be evaluated individually. The emission level $L_{m,E}$ is described by:

$$L_{m,E} = L_m^{25} + D_v + D_{Stg} + D_E \ , \tag{2}$$

where $L_m^{25}$ stands for medium level (25), $D_v$ for maximum speed correction, $D_{Stg}$ for slope correction and $D_E$ for correction of mirror sound sources. The emission position is assumed to be in the middle of the section at a height of 0.5 m above the centreline. The medium level $L_m^{25}$ refers to a horizontal distance of 25 metres, a road surface of unribbed mastic asphalt, a maximum permissible speed of 100 km/h, an incline or slope of not more than 5 % and a free propagation of sound (Bundesministerium für Verkehr, 1990). This results in the following equation:

$$L_m^{25} = 37{,}3 + 10\log(M\,(1 + 0{,}082\,p)) \ , \tag{3}$$

where $M$ stands for average number of vehicles per hour and $p$ for proportion of trucks. The averaging level $L_m^{25}$ of (3) is corrected by additional parameters mentioned in (2); however, these are neglected for faster computing times.

Since the sections do not necessarily have the same length, the emission level must be adjusted by factor $D_l$:

$$D_l = 10\log(l) \ , \tag{4}$$

where $l$ is section's length. It is also important for the calculations to acknowledge the adjustment for distance and air absorption. This component is required to consider the changes to the noise level due to different distances between emission and immission point. The distance and air absorption factor $D_s$ is defined as follows:

$$D_s = 11,2 - 20\log(s) - \frac{s}{200} , \tag{5}$$

where $s$ stands for distance between emission and immission point in metres. Finally, the components $D_{BM}$ and $D_B$, which are included in (1) are defined. $D_{BM}$ describes the ground and meteorology attenuation. In order to keep the computational effort within limits, this factor is ignored. For determining whether a structural condition is required in view of sound insulation, it is nevertheless necessary to include the level change due to topographical and structural conditions $D_B$. Due to simplifications this factor can be described as the negative shield dimension $D_z$, which results in the following:

$$D_B = -D_z = -10\log(3 + 80\,z\,K_w) , \tag{6}$$

where $z$ stands for the shielding factor and $K_w$ for weather correction to compensate the noise beam deflection.

In order to complete the noise calculation for an immission point, the noise values calculated with (1) can be logarithmically added. The sections can be summarised with:

$$L_m = 10\log\sum_i 10^{0,1\,L_{m,i}} , \tag{7}$$

where $L_m$ stands for medium noise level of the complete road and $L_{m,i}$ for that of segment $i$.

Since this plugin is only intended for noise calculation in the early design stages some factors are neglected. Thus a minimisation of the computing effort is achieved and irrelevant precision can be bypassed. This makes it possible to determine the medium level at the place of immission for each single location, alongside the RLS-90 norm. To display the computed results as isophones on the table, a grid calculation is needed. The programming aspects to do so are described in the following section.

## 3 Implementation

Figure 2 shows a brief overview of the tool structure and its functionality. At first, every user has the option to pick a specific region via OpenStreetMaps from which the existing buildings are loaded. An alignment can be drawn with the functions of the interactive alignment tool, presented on the 30$^{th}$ Bauinformatik Forum (Markič et al., 2018).

Subsequently, the sketch data is passed to the developed Noise-Plugin. A noise simulation is started and illustrated with isophones upon the CDP-table. On the basis of the created noise image, the user has the option to place a block on the table to denote the position and orientation of the noise barrier. Through placement of blocks calculations are restarted and the noise expansion picture updated. After finding the best position for the noise barrier, the data can be saved in an Industry Foundation Classes (IFC)-file. It is also possible to create a screenshot of the noise image, to gain a stable reference for future validation and documentation (Pfitzner, 2019).

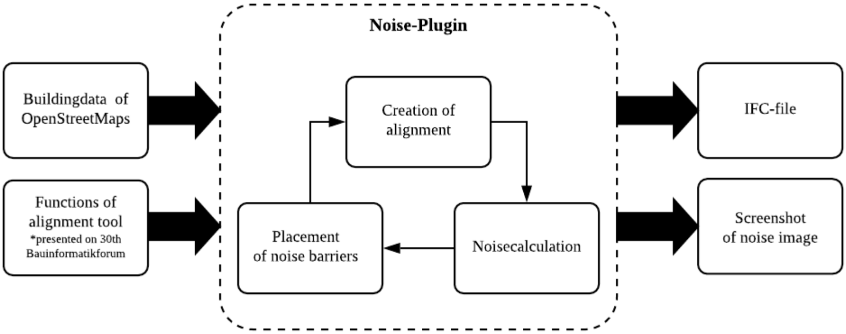



**Figure 2:** Process of the noise tool

The plug-in was designed following the principles of object-oriented programming and an overview of the classes is shown in Figure 3. The *NoiseSimulation* class builds the parent class of all sub-classes. This class only appears in the main program. The sub-classes are all marked with the keyword Noise-, to assign a certain affiliation.

The members of the *NoiseSimulation* class are *Truckshare, AverageCarPerHour, DayMode, BuildingMode*. *Truckshare* describes the truck-share of a single simulation and is declared as a double value. *AverageCarPerHour* is the so-called *durchschnittlicher täglicher Verkehr (DTV)*, which describes the daily average traffic considered in one year. *DayMode* and *BuildingMode* are both booleans and are responsible for switching between day and night mode and deciding whether buildings are included in the calculations or not.

On part of the sub-classes generates the NoiseRender class. This class is responsible for all visible content. It includes drawing the axis, buildings and the coloured noise areas. The class inherits all information from the super class *NoiseSimulation* and passes the data to its sub-classes. To achieve this process we had to create enumerations, where objects like *NoisePoint* and *NoiseCurveSegment* can be saved. Those classes are not only meant for keeping their 2D locations, but also to simplify the intricate computations, pass further information and get better performance.

All calculations are done by the *NoiseSimulation* class. At first all members need to be initialised. Therefore a grid of *NoisePoints* is created. The distance between each point is 10 meter, which turned out to be an optimum between precision and performance, furthermore it's also used within calculations of the Bavarian noise maps by the Bayrisches Landesamt für Umwelt (2013).

Before running the simulation it's important to divide the complete alignment in single curve segments. The length of each segment is depending on the location of the nearest noise point.

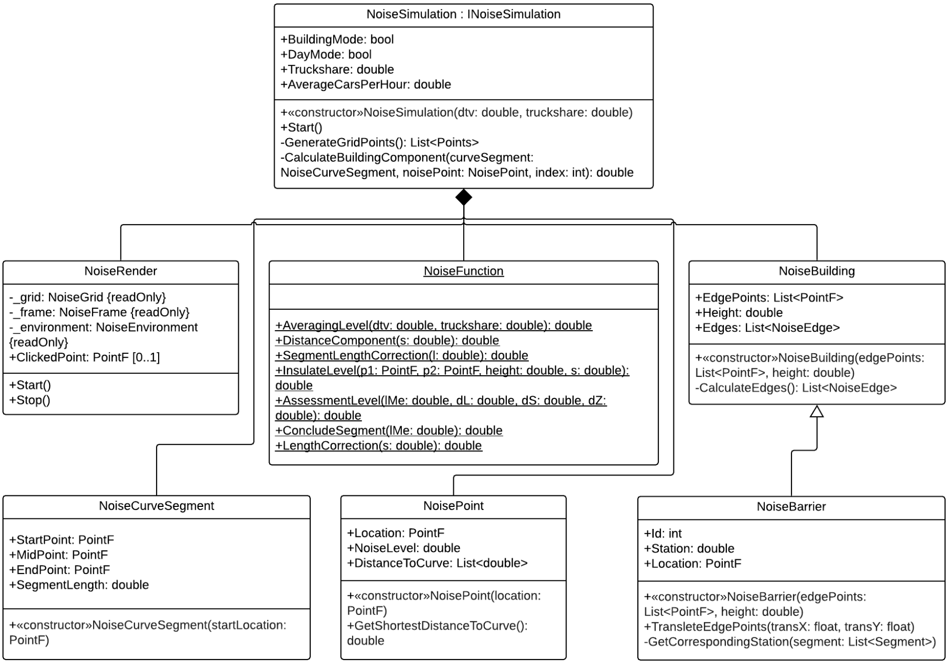



**Figure 3:** UML-diagram of the noise tool

After creating a grid full of *NoisePoints* and splitting the alignment in several parts, computations can be started. The slicing method is applied to each point. The noise level can be computed with the help of the static classes *NoiseFunction* and *AuxiliaryFunction*, which are mostly collections of formulas.

The simulation also provides the possibility to read in building data from Open-Street-Maps. This can be done with the *NoiseBuilding* class. Each building has a certain amount of edges and edge points, which are saved in separated lists, and also defined by their individual classes.

The *NoiseBarrier* class derives from the *NoiseBuilding* class and can additionally be located by its station along the alignment. This is especially important for the later IFC-saving part. Taken a closer look each noise barrier object gets stored in an IFC-file, with the help of an *IfcExport* class. This class associates the geometry of noise walls using *IfcLinearPlacement* and *IfcWall*. This is carried out with the help of the IFC Engine dynamic link library, which provides a collection of external saving methods (RDF Ltd., 2019).

The buildings and barriers are also included in the noise calculations. By placing a block on the CDP-table all data gets passed to a newly created noise barrier object, which will be added to a list of noise barriers. After that a new noise calculation, including the recently added buildings, will be started. The calculated data gets immediately transferred to the

*NoiseRender* class, which results in a noticeable change of the noise image within a second.

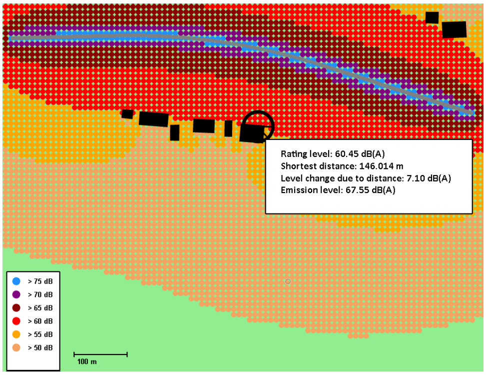
## 4 Verification

For validation purposes, we implemented the possibility to check the noise data of a certain point. By picking one of the grid points a pop-up window with all the information displays. This makes it much easier to control the results at a specific location.

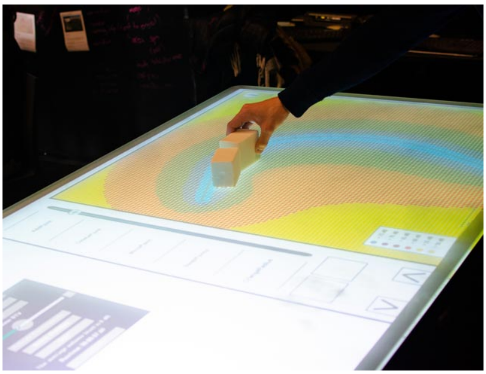
According to the immission location of an example within the RLS-90 standard, a measuring point at a distance of approximately 146 m was selected Bundesministerium für Verkehr (1990). The assessment level on the route is 60.6 decibel(dB)(A), the level change due to all distance factors is 7.1 dB(A) and the emission level is 67.7 dB(A).

The small deviations to the values of the RLS-90 sample calculation are to be justified with the fact that it is difficult to both select a point with exactly the same distance and redraw the road precisely. Nevertheless, the results showed a highly satisfactory agreement.

To get further validation, we also took a comparison to the Bavarian noise maps. The generated noise images also revealed a remarkable rate of compliance (see Figure 4).




(a) Comparison to RLS-90



(b) Intuitive planning process

**Figure 4:** Interactive Noise Tool

## 5 Conclusions

In our study we present an interactive design tool for noise simulations. The main idea of creating this tool was to simplify and optimise the noise calculation process. With the help of the plug-in it is now possible to obtain good results in a timely manner, together with having an intuitive interaction with the simulation tool.

The tool offers both the possibility to create a noise image along a traffic route as well as enables an intuitive placing process for noise barriers, as you can see in Figure 4. This arranges unrestricted possibilities of free positioning and planning. Overall it not only provides an overview of the noise situation along an alignment, but also the influence of a noise protection measure. At the same time, the position of the noise barrier can be stored in an IFC-file along the alignment designed using Markič et al. (2018).

There are two options for further development. First, a screenshot can be created to show the influence of the noise barrier and the resulting noise image as a deduction between before and after to further support the designer. Second, additional design norms could be incorporated to support other design projects.

### Acknowledgements

This paper is a spin off of the Bachelor Thesis written by the first author. He would like to thank the SSF Ingenieure AG for financing this publication and OBERMEYER Planen+Beraten GmbH for helpful comments and guidance during the writing.

## References

Bayrisches Landesamt für Umwelt (2013), 'Straßenverkehrslaerm bayern', `https://www.lfu.bayern.de/laerm/eg_umgebungslaermrichtlinie/kartierung/index.htm`. state: 25.05.2019.

Bundesministerium für Verkehr (1990), *Richtlinien für den Lärmschutz an Straßen RLS-90*, Forschungsgesellschaft für Strassen-und Verkehrswesen. [in German].

Häupl, P., Homann, M., Kölzow, C., Riese, O., Maas, A., Höfker, G. and Christian, N. (2017), *Lehrbuch der Bauphysik: Schall-Wärme-Feuchte-Licht-Brand-Klima*, Springer-Verlag. [in German].

Markič, Š., Schlenger, J. and Bratoev, I. (2018), Tangible alignment design, *in* 'Proc. of the 30th Forum Bauinformatik', Weimar, Germany.

NALS/NATG (2009), *DIN 1320, Akustik*. [in German].

Pfitzner, F. (2019), Intuitiver Lärmschutzwandentwurf entlang einer Trasse in frühen Entwurfsphasen, Bachelor's thesis, Technische Universität München. [in German].

RDF Ltd. (2019), 'IFC-Engine Documentation', `http://rdf.bg/ifcdoc/CS64/sdaiOpenModelBN.html`. state: 30.05.2019.

Schubert, G. (2012), Early Design Support: Interaktive Simulationen in frühen Entwurfsphasen., PhD thesis, Technische Universität München, Lehrstuhl Architekturinformatik. [in German].

Schubert, G. and Petzold, F. (2017), 'Visual programming meets tangible interfaces - generating city simulations for decision support in early design stages'.