SOFTWARE FOR CALCULATION OF NOISE MAPS IMPLEMENTED ON SUPERCOMPUTER

ANDRZEJ CZYŻEWSKI AND MACIEJ SZCZODRAK

Multimedia Systems Department,
Gdansk University of Technology,
Narutowicza 11/12, 80-233 Gdansk, Poland
ac@pg.gda.pl, szczodry@sound.eti.pg.gda.pl

(Received 15 February 2009; revised manuscript received 9 June 2009)

Abstract: Investigation results relevant to the implementation of algorithms for calculation of noise maps are presented in this paper. The aim of implementing the algorithms on a computer cluster is explained. Selected implementation details of the software called *Noise Propagation Model* are described. The software interaction with the data acquisition system is presented. Noise maps obtained using the described software are presented. A comparison between the outcomes of the implemented models and the simulation results of a commercial program is presented. An analysis of the computation efficiency is described. A discussion concerning dynamic presentation of noise maps is also presented.

Keywords: noise map, supercomputer, road noise

1. Introduction

It is required under the European legislation the noise mapping is presented to the public [1]. The way of dissemination is not determined, however, the act states that the most appropriate information channels should be selected. The dynamic development of information technologies which can be observed in recent years, mainly related to the Internet, wireless communication and multimedia computers, enhances the opportunities of applying new technologies in the field of widespread noise hazard assessment. Many efforts have been made by numerous groups to develop noise mapping solutions which has resulted in the development of specialized software. Such systems are based on the noise source and propagation modeling, and they have been applied in most of the large European cities.

The European Commission Working Group Assessment of Exposure to Noise recommends that the data used to assess sound emissions and thereby to carry out strategic noise mapping should reflect the average calculated over the continuous period of twelve months of a relevant calendar year. The map prepared for such input data is very general and does not provide detailed information about the real noise, for example in the last 24 hours. Therefore, a concept of the dynamic noise maps has appeared. Such maps would contribute more to the field of public noise pollution awareness than their strategic counterparts defined by the European Directive 2002/49/EC. That is because the dynamic maps, being regularly updated and based on measured and accurate data, present comprehensive information on the acoustic climate in a given area. The measured data comes from the system of multimedia noise monitoring which has been developed in the Multimedia Department of the Gdansk University of Technology [2, 3]. The data is acquired through a grid of monitoring stations equipped with sensors deployed in significant locations in the city. This data contains noise source parameters required by the numerical model.

The new quality of the frequently updated noise maps, obtained through employing numerical methods, requires that the computation speed problem should be solved. The dynamic noise map is used to present the sound level distribution in a given moment of time. It would take a very long time to calculate such a map for a city area using an ordinary personal computer. This time may exceed the period within which the map should be updated and in consequence render the originated map invalid. A reduction of the computation time requires the application of multiprocessor computers. Most of them are running a Unix-like OS. Commercially available computer applications designed for the purpose of noise mapping usually work with the MS Windows operating system. Having the above problems in mind, the authors needed to develop their own source code of the software for the acoustic field distribution computation and to use open source programming libraries for this purpose. The algorithms for calculation of noise maps were implemented in a parallel programming environment on a cluster-type supercomputer. Thus, it is possible to generate noise maps in a reasonable time and publish regular updates in the Internet.

The results of the investigation of the implemented software for calculation of the noise level distribution in urban areas are presented in this article. The discussed outcomes include the dependence of the calculation time and the number of the applied cores, and a comparison with a commercial application. At the beginning of the paper some implementation details of the software are presented, including a description of the methods used for modeling and the connection between the data gathering system and the software.

2. Numerical solution methodology

The noise map computation software implemented on a computer cluster is a part of a complex solution designed for environment monitoring in cities, called the Multimedia Noise Monitoring System. The system consists of many autonomous, universal monitoring stations, a server which processes and stores the data, a supercomputer which calculates the noise map and a web server which presents the noise map.

Currently, the system gathers information about the road traffic which provides the input data for the noise source model, since in most cases the road traffic is the prevailing source of acoustic climate disturbance. The system is based on a grid of noise monitoring stations. These devices comprise a miniature, industrial PC and a set of sensors to acquire the sound pressure level, along with the associated traffic parameters. Using wireless communication, the data acquired by the grid in question is transferred to the system database at regular intervals.

Each sensor set includes obligatorily a microphone and a camera. The method for acquiring traffic data is based on an analysis of the camera video stream. The main aim of the traffic monitoring is to provide the processed data including noise source parameters. Sophisticated algorithms including Gaussian Mixtures are used to extract the number of vehicles passing by, and their velocity [4, 5]. Vehicles are classified into the desired category groups (mainly according to the utilized source model) also by the traffic monitoring element.

The monitoring stations send the analysis results to a database within a period of one minute. Moreover, the measured quantities, averaged for a period of 1 hour, are collected in order to match the input and the output of the numerical road noise source model. When a request for the noise map update is made, the software gets the current traffic parameters from the database to the road noise model and the computation process starts.

The numerical computation of the noise level generated by the traffic requires that one of the source models is used. In the European Union, the recommended road source model for EU member states which do not have their own model developed is NMPB-96 [6]. In 2004, the Harmonoise project was completed and one of its results was the road noise source model conception. The Harmonoise model [7, 8] was intended to be the one model for European Union state members, and it was designed to replace all different European models. This model uses detailed input data and all calculations are made in 1/3 octave bands. Moreover, the sound emission and propagation are completely separated. It assumes that two separate models for vehicle and traffic have to be distinguished to estimate the noise emission from a linear source representing a road. The vehicle model, describing the sound power of a single moving vehicle, uses the velocity as the input data and returns the sound power output for a specific vehicle type. Each vehicle is represented by 2 noise sources located at different heights. The traffic model, combining the noise emission of numerous vehicles to calculate the sound power per one-meter length of the linear source, provides a statistical description of the sound power output of the total traffic flow. The Harmonoise model assumes a division of road vehicles into categories according to their weight and number of axles. The input data concerning traffic has to be provided in 3 categories (for light, medium and heavy vehicles). Corrections can be applied for different pavement types, tires, road topography (slopes), traffic lights, and source directivity.

Based on the sound power output for a single moving vehicle ($L_{W,m,i}$), the average vehicle speed and the traffic flow, the total sound power output $L'_{W,m,i}$ of each different source height on a unit length road section, in the i^{th} 1/3 octave band is defined by:

$$L'_{W,m,i} = L_{W,m,i} + 10\log\left(\frac{Q_m v_0}{1000 Q_0 v_{\text{eq},m}}\right), \tag{1}$$

where

v_0 – the reference vehicle speed (1 km/h),
$v_{\text{eq},m}$ – the equivalent vehicle speed for vehicle category m [km/h],
Q_0 – the reference traffic flow (1 h^{-1}),
Q_m – the traffic flow for vehicle category m [h^{-1}].

The total sound power output of a unit length road section is obtained by summation over the different vehicle categories, given by:

$$L'_{W,i} = 10\log\sum_m 10^{0.1 L'_{W,m,i}}. \tag{2}$$

The main engine of the discussed noise mapping software is the propagation model which employs the acoustic ray tracing method [9, 10]. The propagation model computes the total sound level in a grid of points which are called receivers. The propagation method [11, 12] describes the attenuation between each source point and a receiver point. In a real atmosphere, the sound propagation is affected by a number of factors including absorption of sound in air, non-uniformity of the propagation medium due to meteorological conditions, and interaction with the absorbing ground and solid obstacles (such as barriers) [13]. The general formula representing the sound level at a certain point is given by:

$$L_p = L_w - 20\log(r) - 11 + D - A_{\text{abs}} - A_E \quad [\text{dB}], \tag{3}$$

where

L_p – sound pressure level [dB] (ref. to $2\cdot 10^{-5}$ Pa),
L_w – sound power level [dB] (ref. to 10^{-12} W),
r – the distance from the source to the receiver [m],
D – directivity index [dB],
A_{abs} – atmospheric absorption [dB],
A_E – excess attenuation [dB].

The total excess attenuation A_E is a combination of all propagation factors, that is: the meteorological conditions, influence of the ground, vegetation, other miscellaneous effects.

The algorithm uses a concept of sound propagation paths representing schematic, straight-line tracks of sound waves a number of factors affects. Point to point (from the point source to the point receiver) sound propagation paths are obtained by a segmentation linear source, resulting in mutually incoherent point sources. The short-term, equivalent sound pressure level $L_{\text{eq1h},i}$ at a certain receiver position is calculated by summation over a number of point-to-point contributions from N propagation paths, according to:

$$L_{\mathrm{eq1h},i} = 10\log\sum_{n=1}^{N} 10^{L_{\mathrm{eq1h},i,n}/10}, \tag{4}$$

where $L_{\mathrm{eq1h},i,n}$ – the short-term, equivalent sound pressure level caused by source segment n (represented by a point source with source power output L_W).

As has been mentioned, the road source model and the propagation model were implemented using some free programming tools and open-source libraries. Supplementary libraries were used: CGAL [14] for geometry primitives, Tardem [15] for importing geographical data, PointToPoint [16] for calculating sound attenuation. Both the road noise model and the propagation model were implemented as one standalone application in the C++ programming language.

The hardware employed for computations is a computer cluster installed in the TASK. Academic Computer Center located at the Gdansk University of Technology. The theoretical computational power of the cluster employing 1344 quad core processors reaches 50 TFLOPS. The real efficiency measured in HPL (High Performance LINPACK) test is 38.17 TFLOPS. The computational capacity makes it the fastest computer in Poland and in the region.

Owing to the employed method of computing sound emission level based on emission and propagation it is possible to obtain a noise level at each given point of the area independently from other points. The usage of the MPI programming standard significantly increases the overall software performance, as all available computer cluster cores are equally charged in this case. The algorithm of parallel processing is shown in Figure 1.

The grid of points in which the noise level should be calculated is the most important input data for the software from the point of view of the overall computation time. A specified number of cluster cores participating in the computation represents a hierarhical structure. The master core manages the data flow and the communication within processors. It distributes the work task to slaves and waits for the results. The work task is defined here as a demand for computing the sound level at one point. The computational process stops when all tasks have been processed. The output data obtained in this way present the noise distribution in a specified region and are recorded in the database.

3. Results

This section presents the results of exploiting of the implemented algorithms. The first experiment concerns the noise map generation speed, especially the dependence of the computation time and the number of the engaged cluster cores. The second experiment estimates the whole city noise map update period. The last subsection shows a comparison of the noise map achieved by the implemented algorithms with the map obtained using commercial software.

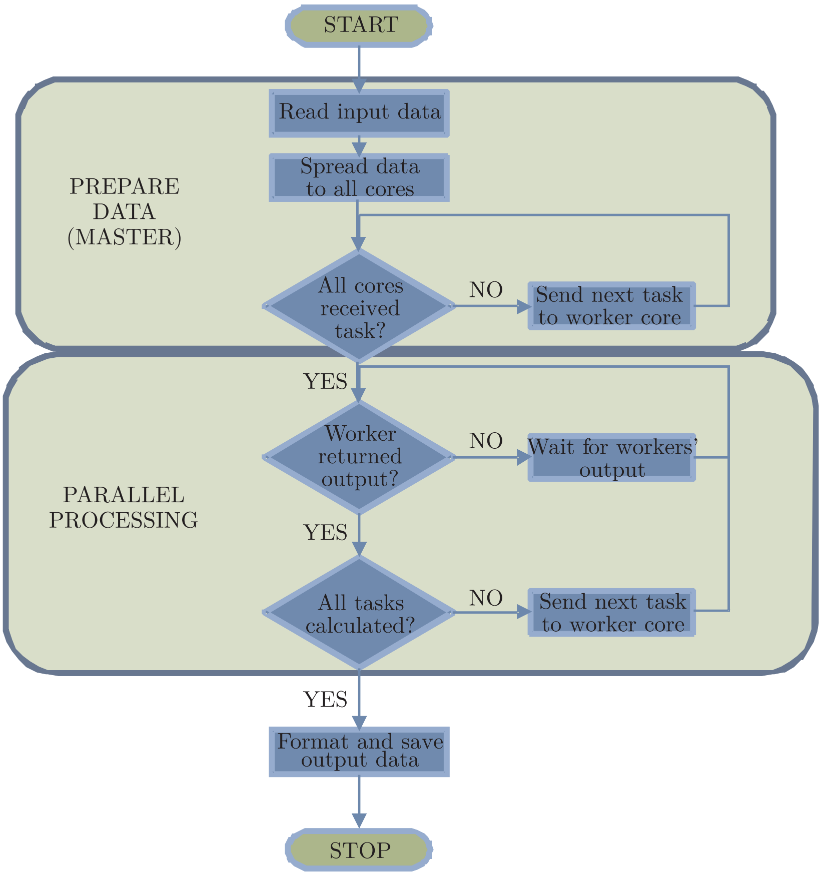

Figure 1. Algorithm of parallel computation of the noise map on a supercomputer

3.1. Dependence of computation time and the number of engaged cluster cores

The calculations were carried out for different fragments of the city map of Gdansk. The dimensions of both the considered areas were 1600×1600 m with a raster of 8×8 m, providing 40 401 points to calculate the sound level. The following main parameters of the propagation model were set: reflections of the 1^{st} order, search ray – 2 000 meters, reflected ray – 1 000 m, the distance between following rays – 2 degrees, and the building sound reflection coefficient – 0.8. The input data for the software consisted of a geometrical description of roads (5 116 road segments) and buildings (91 200 buildings), the traffic volume (fixed for all road segments: 3 000 light vehicles/h, 100 medium heavy vehicles/h and 50 heavy vehicles/h) and the vehicle speed (50 km/h for all categories). All other parameters in the program were set to the default values, *i.e.* stone mastic asphalt pavement type, uninterrupted traffic flow, zero slopes on routes. The ground type for the whole area was set to 20 000 kN·s·m^{-4} in the first case (representing asphalt or concrete) and 80 kN·s·m^{-4} in the second case. The output maps presented in

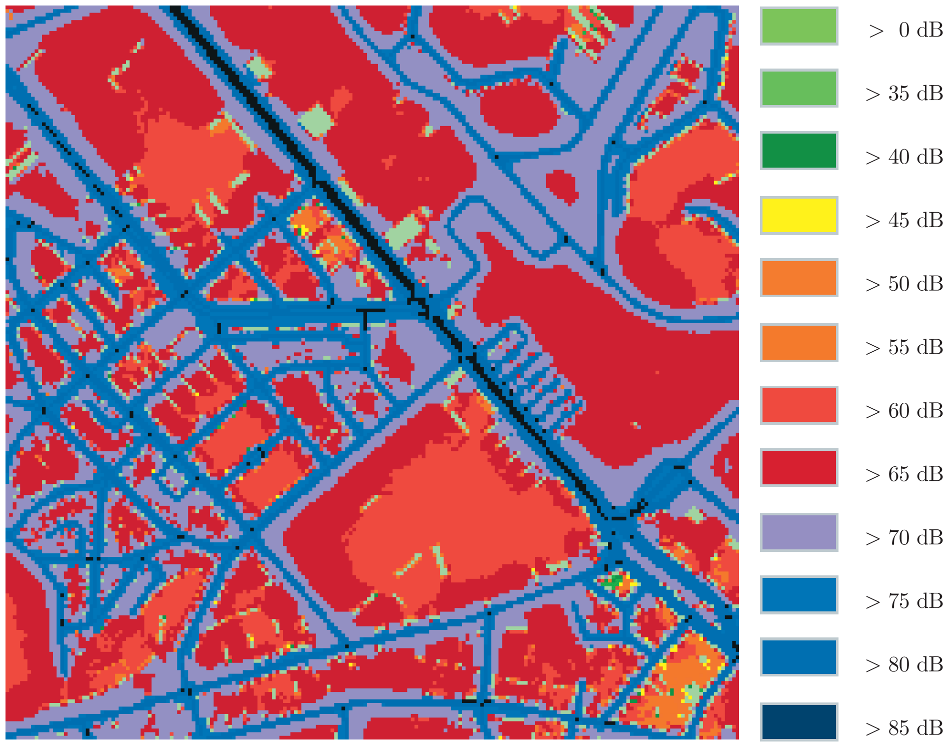

Figure 2. Noise map No 1 for road source, area 1 600 × 1 600 m, raster 8 × 8 m, ground: concrete

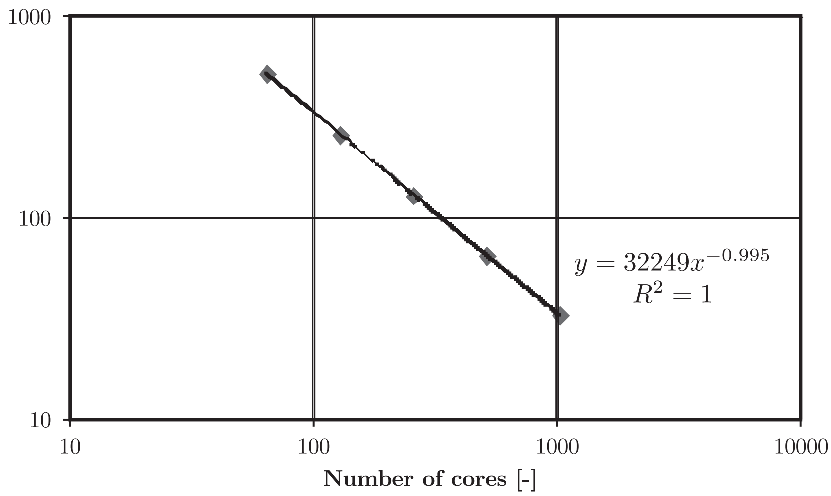

Figure 3. Computation time of noise map No 1

Figure 2 and in Figure 4 are showing sound level $L_{A,Eq}$ averaged for 1 hour. Figures 3 and 5 depict the dependency of the computation time and the number of computer cores for each map.

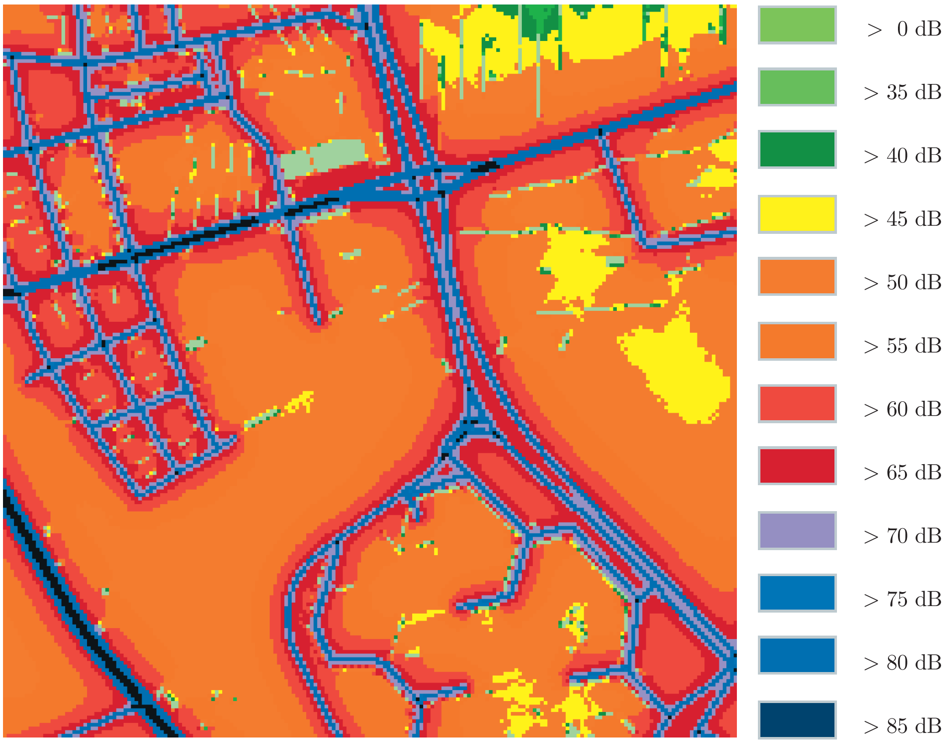

Figure 4. Noise map No 2 for road source, area 1 600 × 1 600 m, raster 8 × 8 m, ground: grass

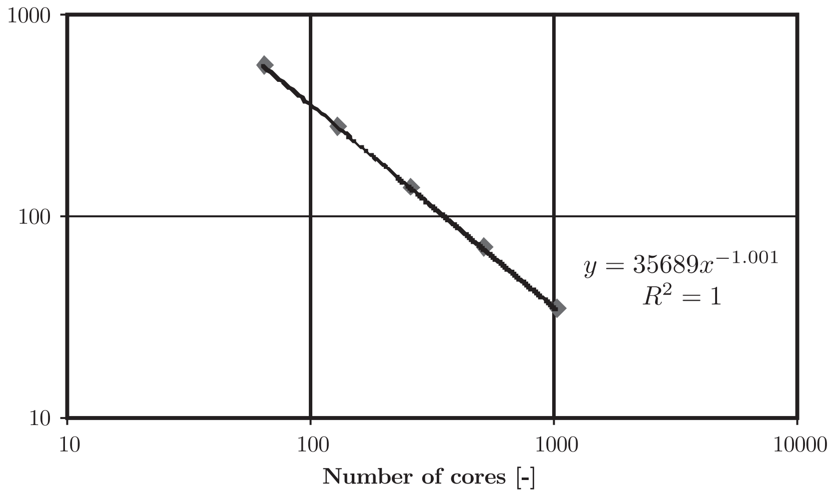

Figure 5. Computation time of noise map No 2

The observations show that the computation time decreases proportionally with the increasing number of the applied cores. Each doubling of the computational power makes the computation last 2 times faster. The dependence is described by equation $y = Ax^{-1}$ in each case, with correlation coefficient 1.

One of the measures of efficiency of the parallel program is a speedup coefficient defined in the following equation:

$$S(n,p) = \frac{T(n,1)}{T(n,p)}, \tag{5}$$

where

n – task size,
p – number of cores,
$T(n,p)$ – time of execution on p cores.

Theoretically, the software should achieve a speedup coefficient close to p. The speedup coefficient can be also determined on the basis of the experiments which were made for checking the dependence of the computation time and the number of the engaged processor cores. The obtained values are shown in Table 1.

Table 1. Computation speedup coefficients for maps No 1 and No 2

Number of cores	Map No 1		Map No 2	
	Computation time [s]	$S(n,p)$	Computation time [s]	$S(n,p)$
64	33249	63.8	30731	63.5
128	16630	127.5	15254	127.9
256	8296	255.5	7634	255.7
512	4150	510.8	3812	512.0
1024	2070	1024.0	1909	1022.4

Because the computation time for 1 core would be very long, the numerator $T(n,1)$ has an estimated value which represents the worst case, according to:

$$T(n,1) = \min(t_{1,64}, t_{1,128}, t_{1,256}, t_{1,512}, t_{1,1024}), \tag{6}$$

where $t_{1,p}$ is the computation time on p cores multiplied by p.

3.2. Estimation of the whole city noise map update period

In the process of creating a dynamic noise map, the key question concerns the computation time of the acoustic field distribution for the whole considered city. The experiment was carried out in order to examine the speed of calculation of the noise map for a large area of $9\,600 \times 9\,600$ m. The spacing of receivers, 8×8 m, resulted in 1 442 401 points to process. All propagation model settings and the input data were the same as in the 2 previous experiments. With 2 032 cores applied, the computation lasted 11.2 hours. The resultant noise map is presented in Figure 6.

The above observation makes it possible to estimate the computation time for the city of Gdansk which covers an area of 265 km^2. If the square shape is assumed, the side length is then 16.3 km. Since the scale between the area sizes is 2.87, the computation time is 32.2 hours. Thus, the map for the whole city can be updated every 48 hours using 2 032 cores. If the period is shorter, more computational power will be required.

It is important to stress that the implemented software is not optimized in any way, nevertheless, it operates in 1/3 octave bands and uses a precise source model the intermediate computations of which consider 3 noise sub-sources for each road segment.

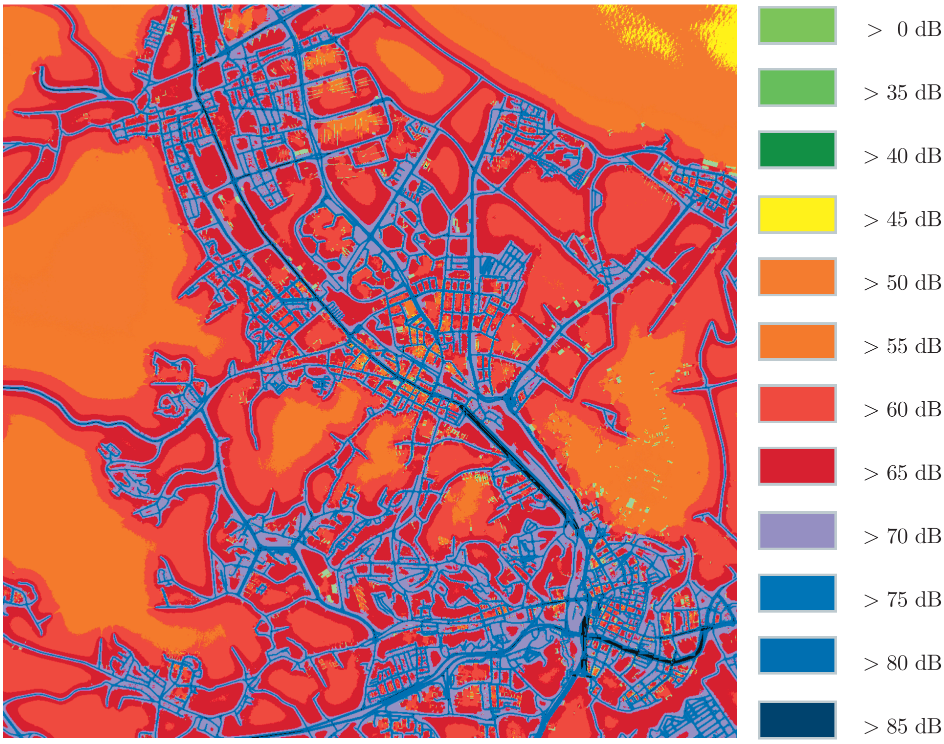

Figure 6. Noise map No 3 for road source, area $9\,600 \times 9\,600$ m, raster 8×8 m, ground: concrete

3.3. *Comparison with commercial software*

The noise maps obtained using the implemented algorithms were compared to those obtained by commercial software CadnaA 3.7 [17]. The software makes it possible to use a variety of models for different noise source types as well as a number of environmental standards. The calculations were made employing the NMPB-96 model as the Harmonoise road noise source model is not yet available in this software. The propagation was calculated according to the ISO 9613 standard [18]. The total noise level at a given point on the city map was derived based on the acoustic ray tracing method, similarly like in our implementations. The maps for the road source are depicted in Figure 7 and Figure 8. The layer containing the municipal infrastructure (roads and

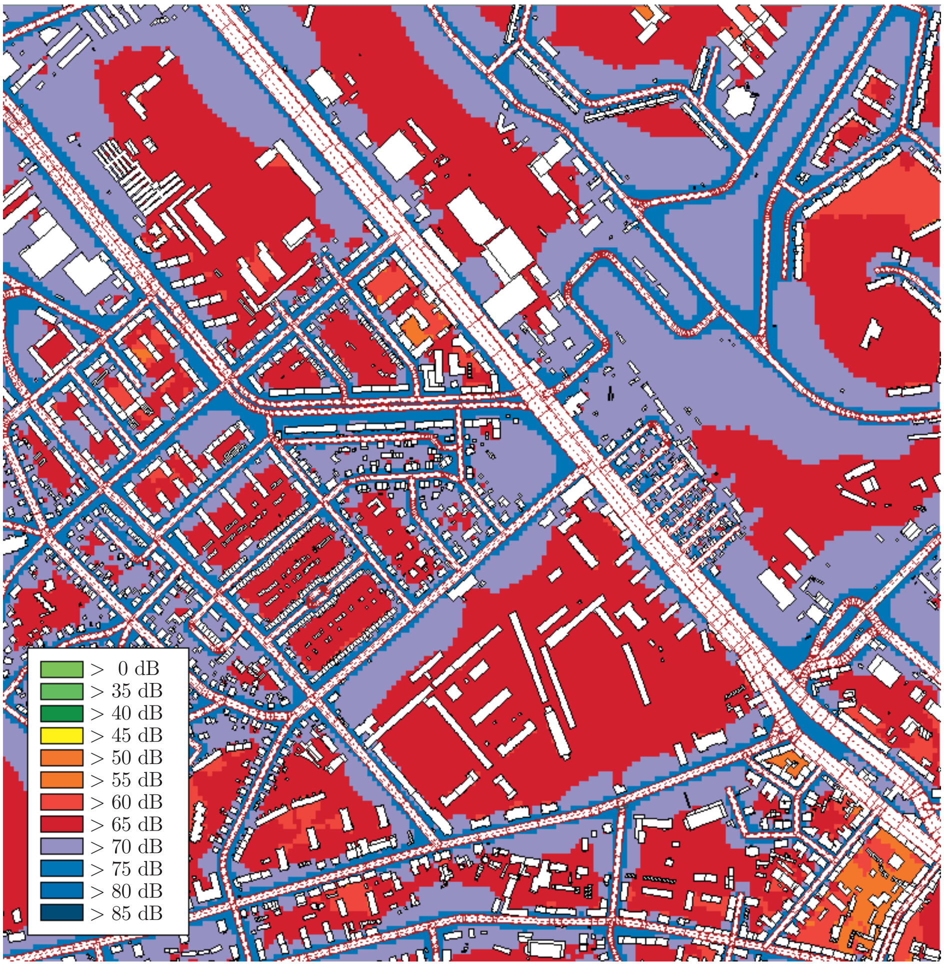

Figure 7. Noise map, CadnaA

buildings) is also shown. The grid of the noise level values calculated on the supercomputer was imported to a commercial program in order to present the results in a standardized way. The main sound propagation model parameters were set in both programs as follows: reflections of 1^{st} order, the search ray – 2 000 m, reflected ray – 1 000 m, the distance between following rays – 2 degrees, and the building sound reflection coefficient – 0.8. The input data for both programs consisted of a geometrical description of roads and buildings (5 116 road segments and 91 200 buildings). The road traffic parameters were fixed for all road segments and consisted of 3 000 light vehicles/h, 100 medium heavy vehicles/h and 50 heavy vehicles/h, cruising at 50 km/h. All other parameters in both programs were set to the default values, *i.e.* stone mastic asphalt pavement, uninterrupted traffic flow, a slope of zero degrees for every road. The ground type for the whole area was set as hard (representing asphalt or concrete) as detailed data were unavailable at that time. Some inconsistencies in the ground type

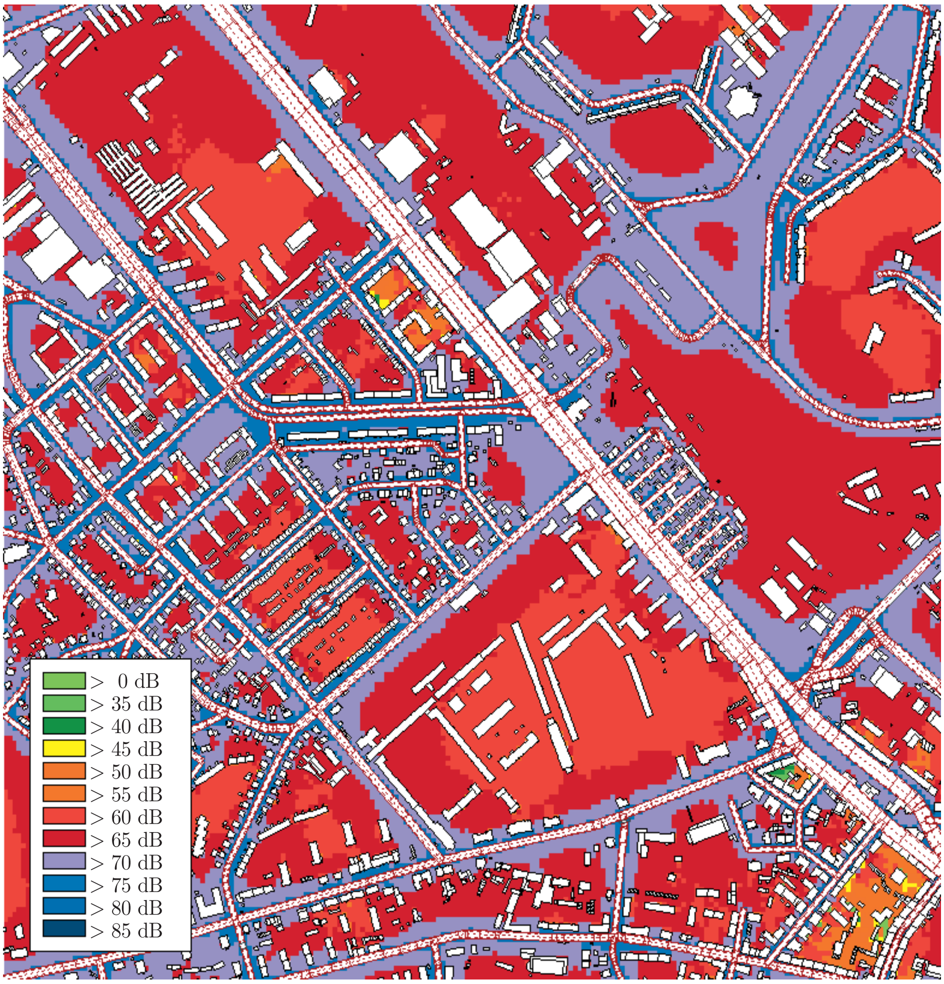

Figure 8. Noise map, algorithms

setup were noticed. The commercial software implements strictly the ISO 9613 standard ground attenuation model, where $G=0$ coefficient means hard ground, and the implemented model uses a more detailed flow resistivity parameter the value of which was set to $20\,000\,\mathrm{kN{\cdot}s{\cdot}m^{-4}}$. The output maps show sound level $L_{A,\mathrm{Eq}}$ averaged for 1 hour. The differences between the sound level values are presented in Figure 9.

The noise level values at each point of the grid were subtracted in order to achieve exact differences, and then quantized to the desired ranges. The commercial software indicates greater results near the road borders and in case of a large distance from the source as can be observed on the difference map. The largest differences between the models reach 4 to 6 dB near high buildings and their aggregation surroundings. The propagation part of the implemented algorithms in this test case overestimated the sound attenuation. It is worth noting

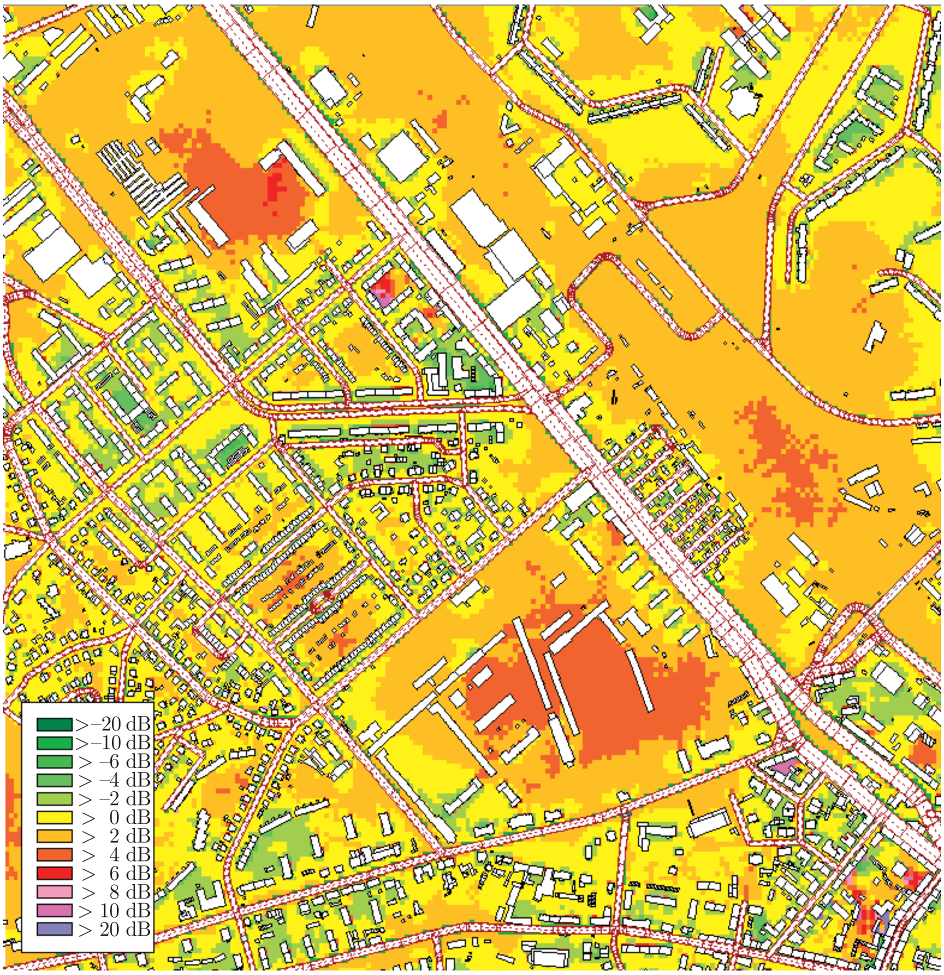

Figure 9. Difference map CadnaA – algorithms

that the commercial program has more parameters to set, *e.g.* concerning the maximum source-receiver distance, the search radius source distance, the search radius receiver distance. Moreover, the noise source models are different, because the Harmonoise model is not yet available in the commercial program and the NMPB-96 model was utilized.

The computation time is also compared. The presented map was calculated within 1 928 s by a supercomputer using 1 016 cores. The commercial program was running for 497 015 s on a 8 core server. The processor in both cases was Intel Xeon Quad-Core 2.33 GHz, 12 MB L3 Cache.

4. Conclusions

The issues presented in this paper constitute a contribution to extend the engineered Multimedia Noise Monitoring System [19] by a possibility of creating dynamic noise maps. The achievement of the intended aim was divided into three stages: partial project, implementation and running of the propagation model and implementation of the noise source model.

The application of a supercomputer in the process of creating a dynamic noise map makes it possible to achieve the result in a reasonable time. The experiments carried out show that a noise map for the whole city of Gdansk can be effectively refreshed every 48 hours. However, the update period in case of a constant number of the input data is dependent on the area size and the computational power.

If the future work focuses on expanding the number of monitoring stations the system will allow for indicating a real noise threat in a city area and help to produce credible noise maps of larger urban areas. The railway noise source model that has been recently implemented in the parallel architecture will be tested. Moreover, the software functionality can be further extended by coordination with a data acquisition system when an extraction of parameters of noise sources originating from trains will be made.

Acknowledgements

This work was supported by the Polish Ministry of Science and Higher Education under Project No. R0201001.

The calculations were performed by computers of the Academic Computer Centre in Gdansk (CI TASK).

References

[1] 2002 *Directive 2002/49/EC of the European Parliament and of the Council of 25 June 2002 Relating to the Assessment and Management of Environmental Noise*, Official Journal of the European Communities 2002.07.18

[2] Szczodrak M, Czyżewski A and Kotus J 2008 *Archives of Acoustics* **33** (4) 77

[3] Czyżewski A, Kotus J, Kostek B and Szczodrak M 2007 *Bezpieczenstwo Pracy* **7–8** 8 (in Polish)

[4] Czyżewski A and Dalka P 2007 *Int. J. Multimedia and Ubiquitous Engineering* **2** (2) 91

[5] Dalka P 2006 *Machine Graphics and Vision* **15** (3/4) 339

[6] CERTU, SETRA, LCPC, CSTB 1997 *Bruit des Infrastructures Routières*, NMPB-Routes-96, Janvier

[7] Jonasson H, Sandberg U, van Blokland G, Ejsmont J, Watts G and Luminari M 2004 *Source Modeling of Road Vehicles, Deliverable 9 of the Harmonoise Project*, Swedish National Testing and Research Institute, Boras

[8] Nota R, Barelds R and van Maercke D 2005 *Engineering Method for Road Traffic and Railway Noise after Validation and Fine-tuning*, Harmonoise WP 3, Technical Report, Paris

[9] Kulowski A 1990 *A Modification of Ray-tracing Acoustics Modeling Method in Rooms*, Zesz. Nauk. Politechniki Gdanskiej, Elektronika, Gdansk (in Polish)

[10] Li K M, Taherzadeh S and Attenborough K 1998 *J. Acoust. Soc. Am.* **104** (4) 2077

[11] Barelds R and Nota R 2002 *Propagation Paths and Reflections*, Harmonoise WP 3, Technical Report, Den Haag

[12] Engel Z 2001 *Environmental Protection against Vibrations and Noise*, PWN, Warsaw, Poland

[13] Embleton T F 1996 *J. Acoust. Soc. Am.* **100** (1) 31

[14] *CGAL, Computational Geometry Algorithms Library*, *http://www.cgal.org*

[15] Tarboton D G 2000 *A Suite of Programs for the Analysis of Digital Elevation Data*, *www.engineering.usu.edu/dtarb/tardem.html*, Utah State University

[16] Van Maercke D 2004 *Programming the Point-to-point Propagation Model, The Harmonoise Engineering Models*, CSTB, Saint Martin d'Hères

[17] 2007 *CadnaA Manual*, DataKustik GmbH, Greifenberg

[18] 1990 *Acoustics – Attenuation of Sound During Propagation Outdoors – Part 2: General Method of Calculation, International Standard ISO 9613–2: 2000*, International Organization for Standardization, Geneva, Switzerland

[19] Kotus J, Czyżewski A and Kostek B 2008 *Noise Control Eng. J.* **56** (6) 497