AN APPLICATION FOR VECTOR-BASED DYNAMIC NOISE MAPS GENERATION

Pawel Kozielecki

Gdansk University of Technology
Multimedia Systems Department
Narutowicza 11/12, 80-952 Gdansk, Poland
`koziol@sound.eti.pg.gda.pl`

Andrzej Czyzewski

Gdansk University of Technology
Multimedia Systems Department
Narutowicza 11/12, 80-952 Gdansk, Poland
`ac@pg.gda.pl`

ABSTRACT

The concept and the developed application for vector-based, dynamic noise maps generation is presented. General features of dynamic noise maps are described. The advantages of dynamic noise maps visualization, utilizing vector graphics and animation techniques, compared to the commonly used software solutions are discussed. The main focus is put on presenting authors' concept of the algorithm for noise contour extraction and shape labeling. The implementation details of the application are presented. Furthermore, quality and fidelity of extracted noise contours are evaluated.

1. INTRODUCTION

Nowadays, environmental noise has become a significant threat to human health, especially in big agglomerations. Long exposure to excessive acoustic disturbances is proven to have a negative effect on human psychics and well-being. In order to assess the discussed threat, numerous technical means has been implemented and installed in the European cities. Such systems mainly focus on measuring noise in key locations of a given agglomeration and visualize its predicted distribution. A popular mean to perform the latter is a noise map. Being a highly suggestive way of presenting information, such chart may significantly contribute to the public noise awareness. In order to achieve that, noise maps must be commonly accessible, precise, descriptive and regularly updated. Acoustic chart of an agglomeration, generated from current data, is referred to as the dynamic noise map.

The solution presented in this paper is an internet browser plugin. It is dedicated to generate dynamic noise maps and to position them geographically. The utilization of vector graphics enabled unmatched control over visual output of the application, together with the control over its performance. The key element of the plugin is an unique mechanism of generating noise maps, thoroughly described in the paper.

2. NOISE MAPS

The legal foundation for undertaking urban noise monitoring is the "Directive of the European Parliament and of the Council relating to the assessment and management of environmental noise" - European Directive 2002/49/EC, published 18/02/2002 [1]. The document obliges European cities with population over 250 000 to determine environmental noise level, according to assessment methods used by all of the Member States. The result is presented as a noise map and published in the Internet to ensure that the public is well-informed about environmental noise and its effects. The ultimate aim of the project is to be able to reduce environmental noise wherever it exceeds acceptable levels. Furthermore, the Directive acts as basis for the development of noise monitoring and reduction solutions.

This section describes the overall concept of measuring and presenting dynamic urban noise. The main focus is on presenting key features of dynamic noise maps in acoustic pollution threat assessment, in comparison to static, long-term ones.

2.1. Strategic Noise Maps

Strategic Noise Map is defined as a map, designed for the global assessment of noise exposure in a given area due to the existence of various noise sources, the map serving overall predictions for such an area. It presents an existing, a previous or a predicted noise situation based on noise indicator computing. These indicators outline noise level, measured for different time intervals. The most common noise indicators include: L_{den} (day-evening-night noise indicator) and L_{day} (day-noise indicator) [1]. Noise distribution, measured or predicted according to the particular noise indicator, creates a separate map. Joint together, these maps comprehensively describe noise threat for a given city. Furthermore, separate strategic noise maps must be made for road-traffic noise, rail-traffic noise, aircraft noise and industrial noise. Maps for other sources may be also added.

However, precisely defined, according to the Directive [1], Strategic Noise Maps do not need to be frequently updated. Maps for agglomerations require revision at least once in 5 years. Moreover, every update of the map requires performing time-consuming and expensive data acquisition procedures. This is true even for predicted maps, provided that input data are not gathered automatically, i.e. by noise measuring terminals [2]. In addition, there is, common among municipal authorities, negligence for civilization diseases resulting from long exposure to excessive noise. Consecutively, most of the contemporary Strategic Noise Maps present static, outdated content. As such, they contribute rather little to the public noise pollution awareness. Hence, there is a strong need to develop an alternative solution - a system to generate and to visualize current noise distribution in urban areas, based on dynamically gathered data [3]. The output of this system is referred to as a dynamic or an online noise map.

2.2. Dynamic Noise Maps

The dynamic noise map is used to present noise in a given time moment. In order to create such map, a great number of points with known noise level is required. For practical reasons, actual sound pressure level is acquired in few key locations of the city, only. The noise distribution in remaining points of the map is determined with the utilization of some computational methods. In most cases the procedure is performed by a dedicated, predictive noise model [4]. Such an application utilizes a wide array of quasi-static (i.e. architectural) and dynamic (i.e. environmental, meteorological and road traffic) data. The latter require regular updates and need to be acquired through set of specialized monitoring stations [2]. Unfortunately, time needed to generate a map on a standard desktop computer is too long to meet the requirements concerning frequent map updates. Therefore, dynamic noise maps are calculated with the utilization of cluster-type supercomputers [5]. The task of determining the noise distribution on a given area is divided among all available processors in the system and summarized as a matrix of noise samples. Such a matrix may be generated separately for different noise indicators and presented to the user in many different forms, i.e. tabular, as a vector or raster graphics.

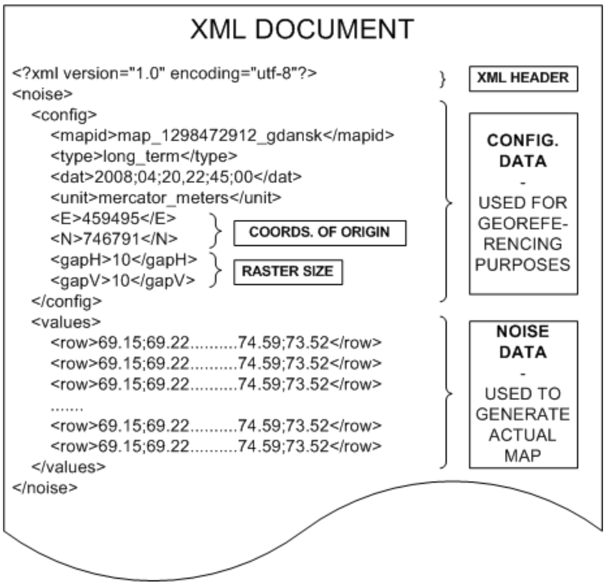
Ensuring regular and frequent updates of the dynamic input data, allowed the system to present credible and informative noise maps. In addition, noise values used to generate past noise chart, may be stored in the system database and used to highlight changes of the acoustic disturbance distribution in a given time interval. Such a system contributes more into the field of noise monitoring than any static map. Moreover, the cost of updates is reduced to the maintenance cost of the noise monitoring stations. In most cases, this cost is similar or less then the cost of performing complete measurements for a Strategic Noise Map.

3. DYNAMIC NOISE MAPS GENERATION UTLIZING VECTOR GRAPHICS

This section outlines key features of the proposed solution. The advantages of utilizing vector graphics to present dynamic noise maps are presented. Furthermore, the authors' conceptual algorithm dedicated to visual output of the system generating is described. An implementation details of the application are presented at the end of the section.

3.1. Principles of the proposed approach

The Multimedia System for Dynamic Urban Noise Monitoring [3] requires specialized application to visualize and to georeference resulting noise maps [6]. The application is implemented as a standard internet browser plugin. Therefore it fulfills the 2002/49/EC Directive with regards of providing common, public access to the generated maps. The main feature distinguishing the application from a common digital maps visualizing tools is its client-side noise maps generation process. Information required to create such map is sent to the plugin as a set of raw, numeric values (stored in a matrix). This data may come from real-life measurements or from a prediction model. Numeric noise map must also contain its geographic coordinates and a scale factor. The structure of discussed data is shown on Fig. 1



```
<?xml version="1.0" encoding="utf-8"?>
<noise>
  <config>
    <mapid>map_1298472912_gdansk</mapid>
    <type>long_term</type>
    <dat>2008;04;20,22;45;00</dat>
    <unit>mercator_meters</unit>
    <E>459495</E>
    <N>746791</N>
    <gapH>10</gapH>
    <gapV>10</gapV>
  </config>
  <values>
    <row>69.15;69.22..........74.59;73.52</row>
    <row>69.15;69.22..........74.59;73.52</row>
    <row>69.15;69.22..........74.59;73.52</row>
    .......
    <row>69.15;69.22..........74.59;73.52</row>
    <row>69.15;69.22..........74.59;73.52</row>
  </values>
</noise>
```

Figure 1 *An overview of the XML message for noise maps generation*

In order to transform received information into graphical form, the application is equipped with specialized algorithms for contour extraction, labeling and optimization, described in detail in Sect. 0. As a result of mentioned contouring routines, a set of vector layers, corresponding to a given noise level (i.e. greater than 85 dB SPL), is created. Each layer consists of numerous shapes, separated and filled in with appropriate color. The fill color is calculated according to the sound level corresponding to a given layer. To add geographic context to the noise map, it is positioned on an underlying city map. This map can be delivered in vector or raster layers. The positioning is done through an internal positioning engine, utilizing cartographic projection appropriate for a given region. i.e. for the city of Gdansk, Poland, “Uklad 2000” [7] cartographic projection is applied for most precise geographic positioning. Alternatively, for the regions without officially recommended cartographic projection, the Universal Transverse Mercator (UTM) [8] can be used. In order to highlight the varying level of noise in a given time interval, the noise map can be animated. Additionally, the application is equipped with some modules to present other media, such as temperature or atmospheric pressure, or discreet objects.

3.2. Reasons for presenting dynamic noise maps in vector format

As is mentioned Sec. 3.1, the actual noise map is delivered to user in the vector format. This constitutes a novel approach, compared to the commonly used solution presenting noise distribution as raster images. Presenting acoustic charts in the proposed way has a multiple justification.

Initially, for a multiple scale images, it is more economic format compared to the raster solutions. Precise visualization of a raster noise map requires numerous sets of raster tiles – a separate set for a given map scale. Upon entering particular map scale, currently displayed tiles are replaced with new ones. The discussed procedure is time consuming, in terms of waiting for all tiles to be downloaded. Additionally, it requires an efficient HTTP server and wide internet connection bandwidth to offer satisfying performance in cases of multiple, simultaneous system accesses. Alternatively, in the proposed solution, a single image is created. It is generated for the largest map scale and gradually generalized whenever downscaling occurs. The operation is performed automatically by the application of the CAD engine. Apart from the initial request for the noise data, the server-side part of the system is not engaged during rescaling operations. It is a vital issue concerning system running costs and the demand for server-side resources.

One of the top priorities of the Multimedia System for Dynamic Urban Noise Monitoring visualization module is to enable unlimited user contribution into the map generation process. This can be done only by transferring the process to a client terminal and delivering the output in a vector format. Three groups of parameters affecting appearance of the map are defined. These are: contour tracing, contour reduction and shape drawing group of procedures. Experimenting with the parameters setup may produce nearly unlimited different visual outputs. The complete list of all parameters and their possible setup is given in Tab. 1.

Table 1. *Parameters utilized in the map generation process*

Parameter Family	Parameter Name	Parameter Function
Contour Tracing	Maximum noise level value	defines noise value bound to the “hottest” color on map
	Minimum noise level value	defines noise value bound to the “coolest” color on map
	Step	defines noise level gap between neighboring map layers; the minimum value is 1 dB SPL
	Blur level	defines a size of an averaging matrix filter applied to the input noise values array; the operation smoothens resulting shapes; is applied before contour tracing
Contour Reduction	Reduction Method	type of reduction applied to decrease the number of points defining contours; possibilities are: simple decimation, arithmetic average, Lagrange interpolation
	Reduction Factor	number of key points to be reduced in a single step of the algorithm
Shape Drawing	Interpolation Method	method applied to connect knots and form a closed shape; possibilities are: straight lines joining, Catmull Splines and Bezier Splines
	Interpolation Precision	required only for Spline interpolation; defines number of subintervals of a single Spline

3.3. The algorithm for vector dynamic noise maps generation

The concept of the algorithm implemented in the application is presented in Fig. 2:

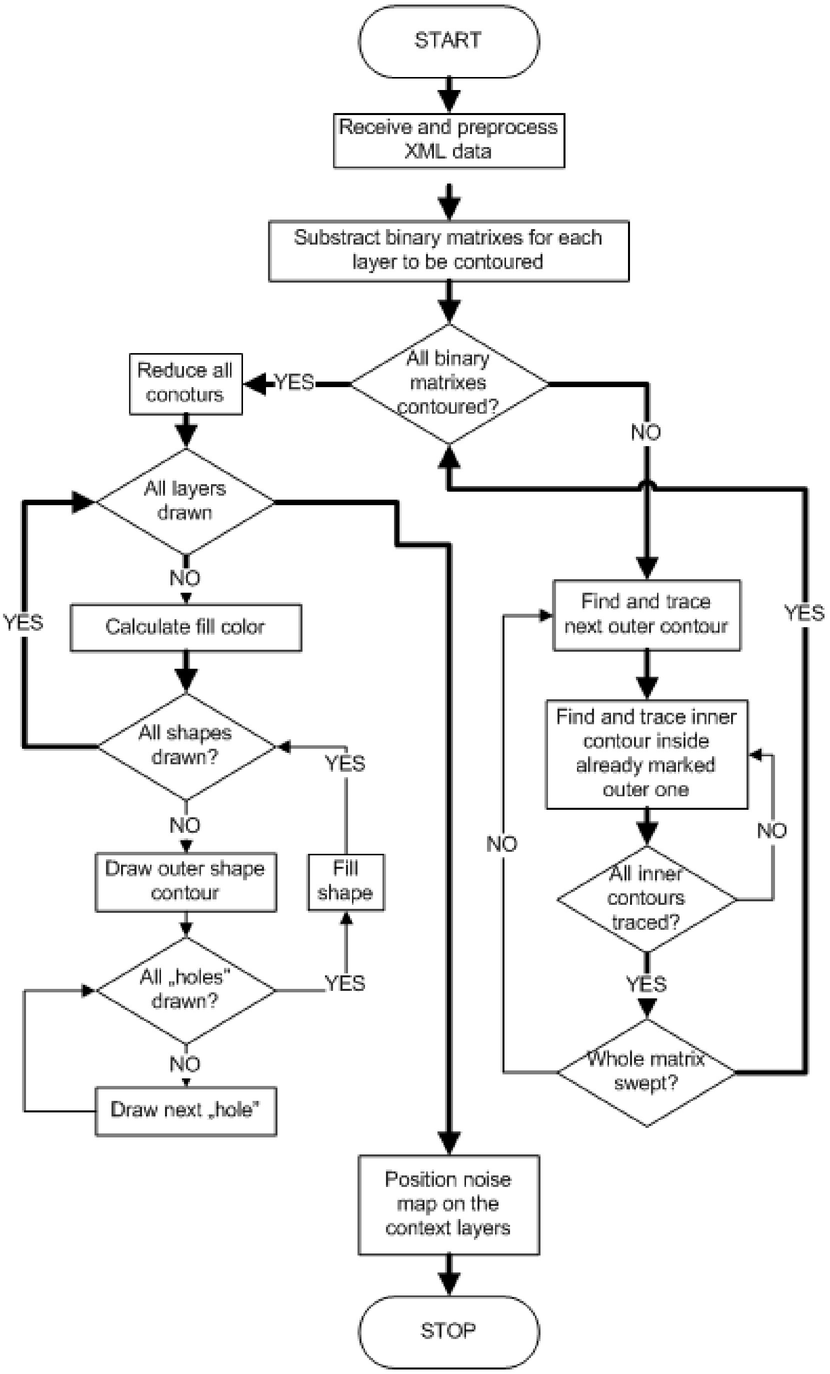


Figure 2. *An outline of the dynamic vector noise map generation algorithm*

The algorithm consists of two main parts: first, implementing noise contour extraction, second, realizing contour reduction and drawing. For user convenience, certain parts of the algorithm, related to the operations of contouring, reduction and drawing, can executed independently, based on previously processed data, i.e.

different reduction procedures may be used on the output of contouring subroutine. Likewise, the noise map may be drawn utilizing all available methods, based on reduced contour descriptions.

As is mentioned Sect. 3.1, data utilized in the noise map generation process are delivered to the application in a raw numeric format. Therefore, some preprocessing (i.e. data cleaning) must be performed. For the purposes of the contouring subroutine, a set of binary matrixes is created for each layer. A cell value is set to "1" if its counterpart in the noise values array is greater or equal to the value assigned to the given noise layer, i.e. for a layer marked "75dB", in the corresponding binary matrix, ones indicate values grater then 75dB. Generated binary matrixes are delivered to the contouring subroutine. The procedure itself is an adaptation of the solution proposed by Chang et. al. [9]. Essentially, it sweeps through the matrix, seeking for the outer contour of a shape, traces it and advances to find all the inner contours inside. The letter can be referred to as the "hole-seeker". When all of the internal and external contours in the given matrix are traced and labeled, the algorithm moves to the next one. Extracted contours are described by their (x,y) coordinates. Though being precise, this form of description is also highly redundant. Initially generated number of points is too large to be efficiently used by dynamic vector shapes drawing algorithms, especially those based on splines. Resulting geometric forms are angular and highly irregular, because distances between neighboring nodes are too small. As a result, before drawing, contours need to be reduced. The reduction method and rate are chosen by the user. Sample shapes, drawn with and without the reduction, are presented in Fig. 3 and Fig. 4.

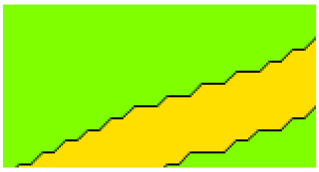


Figure 3. *Shape drawn based on an unreduced noise contour*

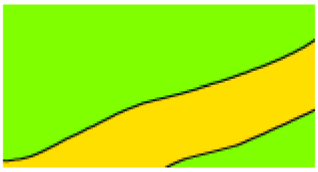


Figure 4. *Shape drawn based on a noise contour with number of nodes reduced 5 times*

For each noise layer an appropriate color is calculated. It is performed according to the chosen color scheme, minimum and maximum noise values selected by user. Available color space between the "coolest" and the "hottest" color in the scheme, is divided among available noise values. Regardless of the nodes joining methods (i.e. straight lines or splines), the shape drawing procedure is essentially the same. Contours are sketched in the order they were traced. After a given outer contour, all its "holes" are drawn. Afterwards, algorithm moves to the next outer contour. The procedure is illustrated on Fig. 5. After the noise map is generated, it is positioned on the context layers. Afterwards the algorithm concludes.

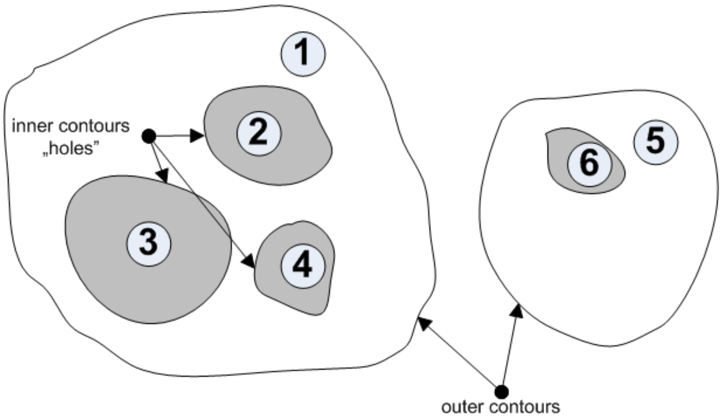


Figure 5. *Noise shapes drawing order*

3.4. Implementation details

The main objective of the Multimedia System for Dynamic Urban Noise Monitoring is to disseminate knowledge about the environmental noise threat in the society. Therefore, an application to visualize dynamic noise maps had to ensure high accessibility standards. These include an ability to access the system regardless of user computer (i.e. Mac or PC), internet browser or Operation System. Therefore, the application has been implemented in the SWF file format using Actionscript® 3.0 programming language. The SWF is an open-source standard for presenting dynamic vector and raster graphics in the Internet [10]. It is currently a dominant solution in its class of application, offering unmatched drawing performance and quality. The discussed platform is also known for highly developed tools to create robust and user-friendly graphical user interfaces (GUI). The application interface is built from custom components, designed to ensure maximum visibility (smallest possible obscured area) of the map and performance. The latest application GUI is shown in Fig. 6.

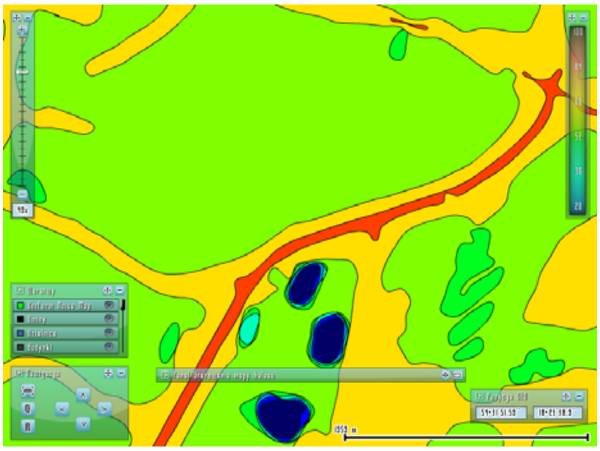


Figure 6. *Sample of the application interface – currently displaying only in Polish language*

The SWF platform offers support for many dynamic data provider solutions, such as AJAX or databases. This is utilized in the procedure of acquiring data from noise map generation, user settings management and many more.

In order to ensure sufficient performance of the application running on a low-end terminals, certain visual quality controllers have been implemented. These include: bitmap caching procedures, intelligent shape masking and generalization, global and local anti-aliasing settings. The first two methods are applied to reduce number of mathematical operations required to move, rotate or rescale map elements. For a given map scale, complicated vector shapes can be cached as a raster representation and, as such, efficiently transformed. Likewise, eliminating elements outside of the visible area of the application significantly contributes to the overall application performance. This is done through an intelligent shape masking. Map elements are also automatically generalized upon rescaling the map. Apart from the discussed methods, global and local anti-aliasing parameters may be adjusted.

4. EXPERIMENTS

This section describes tests performed using the application. The performance of the map generation process is evaluated with focus on time required to execute key sections of the algorithm described in Sect. 0. Afterwards, the accurracy of the generated vector maps, in comparison with raster counterpart, is assessed.

4.1. Map generation performance tests

The tests were performed to evaluate performance of the application and find optimal values of input parameters for the dynamic noise maps generation algorithm. The algorithm was decomposed into its key sections: contour tracing, contour reduction and drawing subroutines. Tests were performed for a map covering 4km x 4km area of uniformly distributed points (10 m vertical and horizontal spacing) of determined sound pressure level. Time required to accomplish each section was measured and averaged for 10 runs. Tests were performed for two parameters: gap between neighboring noise layers (in dB) and size of the blurring filter applied to the raw noise samples values. These factors contribute significantly to the map generation time, level of complexity and fidelity. Other parameters were set constant during the experiments. Maximum noise value was set to 100 dB, minimum to 20 dB, contours were reduced used arithmetic averaging with the factor of 5x and drawn with Bezier Splines with 10x precision. Fig. 7Figure 7 presents time intervals required to accomplish discussed key sections of the noise maps generation algorithm. Test were performed for fixed blurring filter size of 1x. Result acquired for the fixed gap size of 10 and varying blurring filter size are given in the Fig. 8Figure 8. The machine used for experiments was quadruple core (4x3GHz) PC computer with 4 GB RAM memory and installed Microsoft Vista OS.

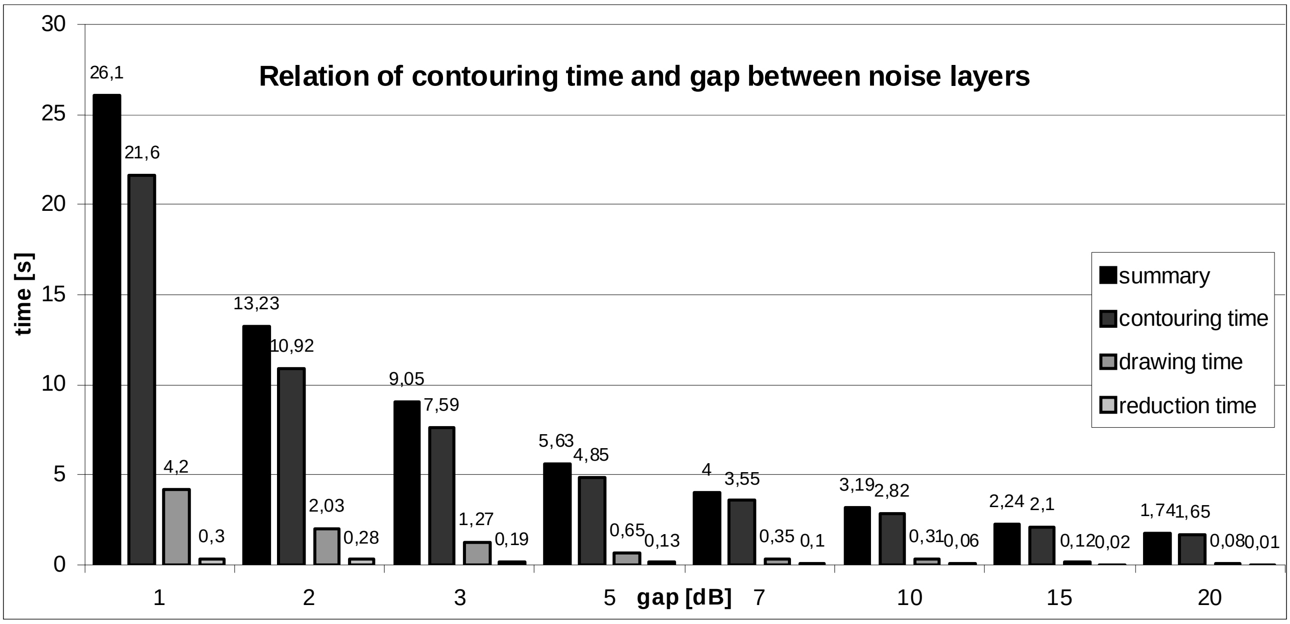


Figure 7. *Completion time for key sections of the noise maps generation algorithm*

As can be observed, the most computationally demanding part of the algorithm is contour tracing subroutine. It is also, potentially, the field for performing algorithmic adjustments in the optimization process. Because the tests were performed only for a part of the target map (approximately ¼ of the whole chart), the recommended size of the gap among noise layers is between 5 and 10 dB. This will allow to produce visually accurate map, keeping its generation time within 20 s. Choosing 3 dB gap will result in a more precise output, however for the price of a significantly longer waiting time. Alternatively, choosing too large gap produces visually deceiving maps. The top boundary for the gap size, allowing for generation of reasonably precise output, is 10dB.

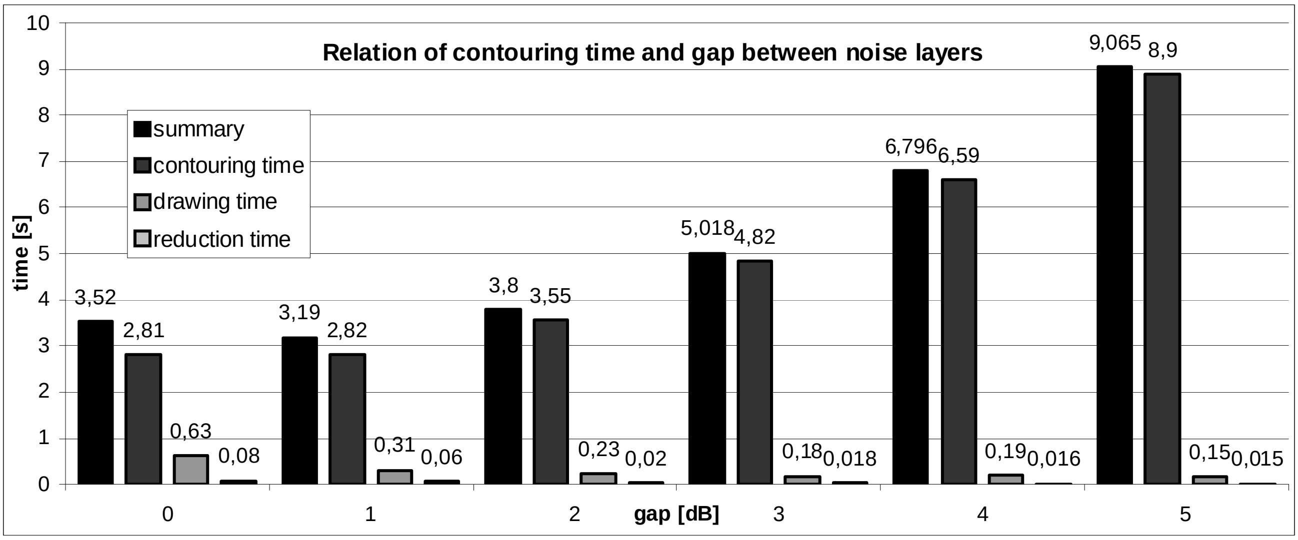


Figure 8. *Completion time for key sections of noise maps generation algorithm*

As can be observed, the complexity of blurring operations does not contribute significantly to the map generation time. The relation between blurring filter size and time required for contouring is nearly linear. The discussed operation is required to prepare the matrix of noise values for the contouring subroutine, i.e. it smoothens the edges and removes possible data corruption. In order to preserve visual precision of the map, an optimal blurring filter size should be chosen between 1x and 2x. This allows achieving smooth contour edges, preserving high level of detail in the map.

Generally, the size of blurring filter of 1x with the gap between noise level layers set to 5dB, should result in a visually precise noise map.

4.2. Map fidelity tests

Map fidelity test were performed to determine optimal parameters settings that would allow generating visually precise map maintaining high application performance. Additionally, the lower boundary on map scale was determined, on which no visual difference can be found between maps created utilizing different curves approximation techniques. This is a significant issue, because the complexity of the map highly depends on the applied drawing method. Figs 9Figure 9, 10, 11, and 12 present maps (scale 1:25000) generated for a given gap between noise layers in comparison to a raster counterpart. Maps were created for 20 dB minimum and 90 dB maximum noise value, and for 3, 5, 7 dB gap.

As can be observed, optimal visual effects can be achieved for a gap value around 5 dB. Greater values of the discussed parameter may result in too general and visually deceiving output. Alternatively, setting gap to less than 3 dB may lead to overdetailed presentation and performance loss. In general, the application produces precise output, provided that input parameters are tuned to the unique features of the given urban noise distribution.

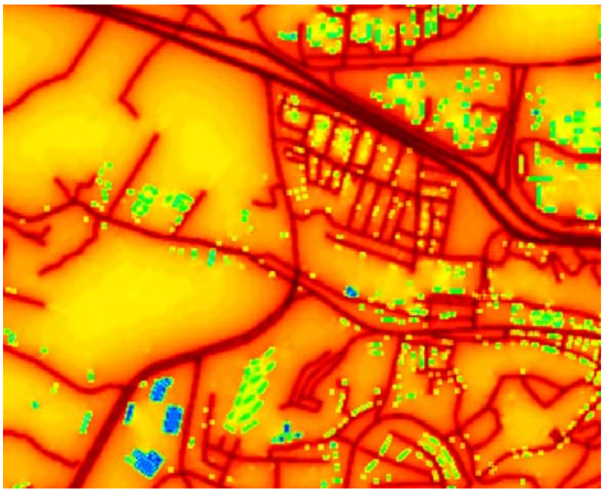


Figure 9. *Dynamic noise map generated as a raster graphic*

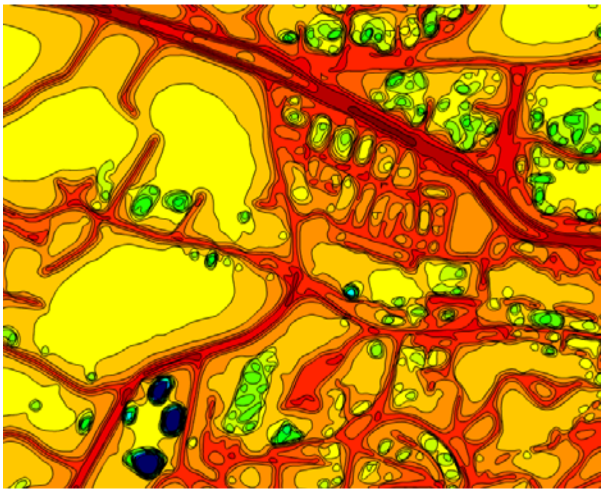


Figure 10. *Dynamic noise map generated for 3dB gap*

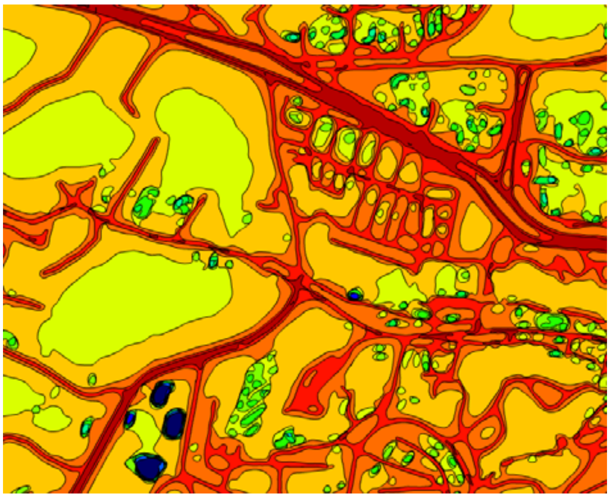


Figure 11. *Dynamic noise map generated for 5dB gap*

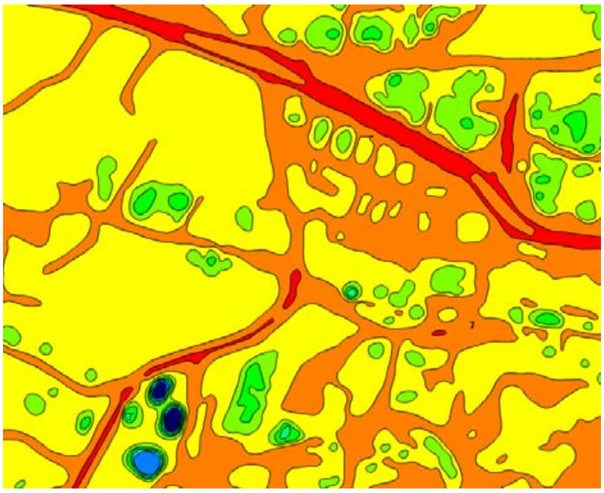


Figure 12. *Dynamic noise map generated for 7dB gap*

The second part of the tests concerned lower determined map scale value, allowing to choose less accurate drawing method without significant difference in visual output. These procedures included Bezier and Catmull Splines and straight lines joining. Both spline-based techniques compose curves of short, straight line segments. However, less accurate (producing more angular and rough shapes), Catmull Spline offers a better general application performance wherever zooming, panning operations are concerned. Fig. 13, 14, 15, and 16 present noise maps, scaled approximately 1:45000 and 1:54000, generated with the utilization of discussed spline-based techniques. A similar comparison, for smaller map scales, is presented on Fig. 17, 18, 19, 20, 21, and 22. The setup also includes maps drawn by a straight line joining of the contour nodes. However, introducing a considerable curve approximation error, it offers optimal application performance and can be applied when the scale of the map is sufficiently small.

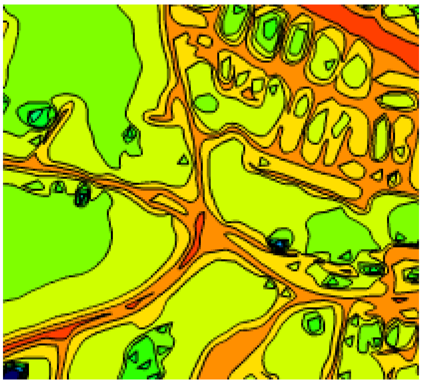


Figure 13. *Dynamic noise map scaled 1:45000 generated utilizing Catmull Spline curve approximation technique*

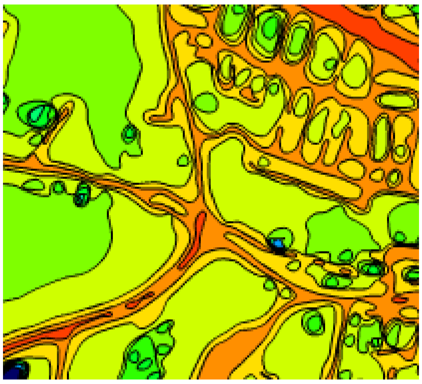


Figure 14. *Dynamic noise map scaled 1:45000 generated utilizing Bezier Spline curve approximation technique*

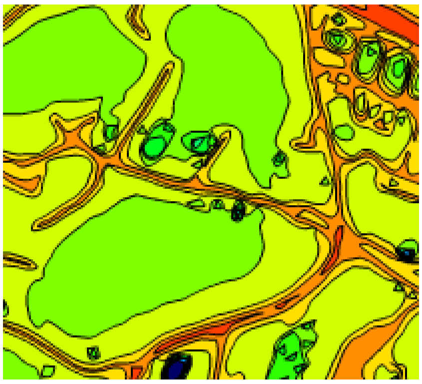


Figure 15. *Dynamic noise map scaled 1:54000 generated utilizing Catmull Spline curve approximation technique*

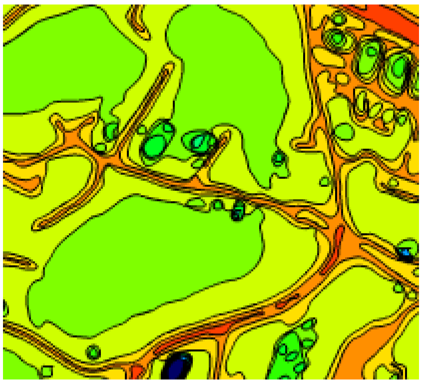


Figure 16. *Dynamic noise map scaled 1:54000 generated utilizing Bezier Spline curve approximation technique*

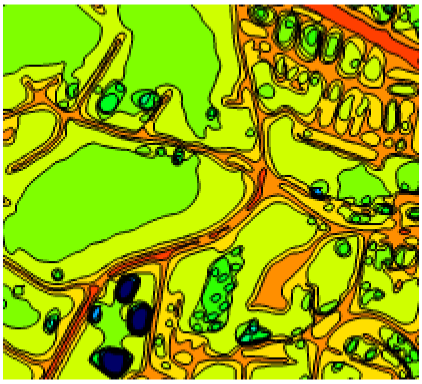


Figure 17. *Dynamic noise map scaled 1:64000 generated utilizing Bezier Spline curve approximation technique*

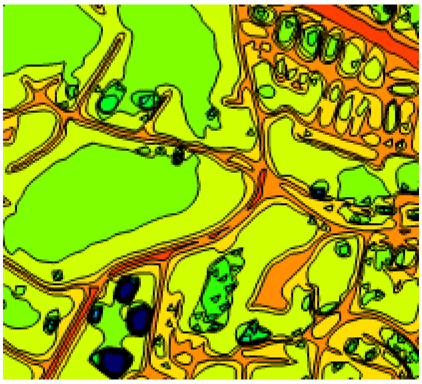


Figure 18. *Dynamic noise map scaled 1:64000 generated utilizing Bezier Spline curve approximation technique*

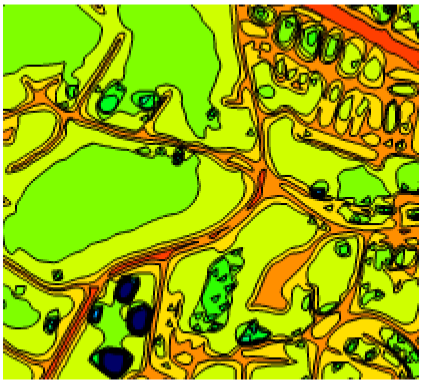


Figure 19. *Dynamic noise map scaled 1:64000 generated by joining contour nodes with straight lines*

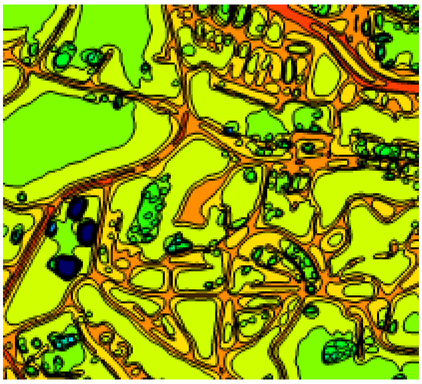


Figure 20. *Dynamic noise map scaled 1:80000 generated utilizing Catmull Spline curve approximation technique*

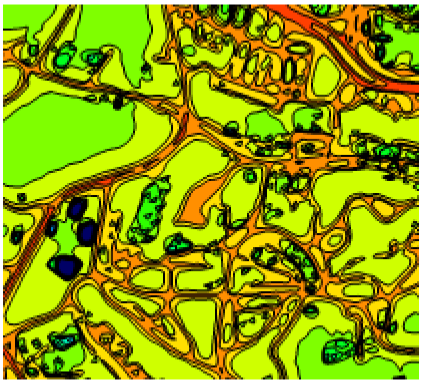


Figure 21. *Dynamic noise map scaled 1:80000 generated utilizing Bezier Spline curve approximation technique*

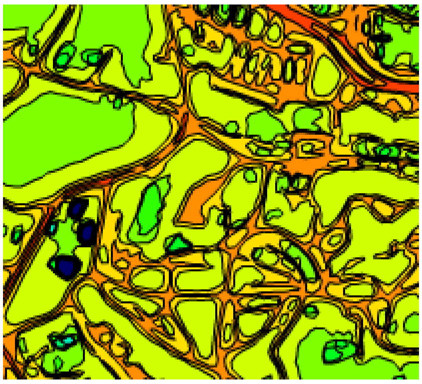


Figure 22. *Dynamic noise map scaled 1:80000 generated by joining contour nodes with straight lines*

As can be observed, for the medium scale map, less accurate curves approximation techniques may be applied. For 1:45000 maps, the application of Catmull Splines results in a slight visual difference, especially for small, circular shapes. The map scaled 1:54000, drawn the same way, exhibit virtually no generalization errors for developed and medium shapes. For small polygons however, visible angularities may still be observed. In general, the bigger scale, the more generalization errors will manifest themselves. Therefore, for maps smaller then 1:45000, in order to ensure a better performance of the application, Catmull Spline approximation technique may be used instead of Bezier Spline one.

Similarly to the previous comparison, the general regularity concerning the map scale applies to the straight line joining as well. Approximation errors are more visually noticeable then ones generated by other discussed techniques. However, for map scales smaller then 1:64000, they are negligibly small. Therefore, when application is run on low-end terminals, straight lines joining may be used to guarantee convenient use of the system.

In conclusion, recommended curve approximation technique relies heavily on the scale of a map. For a large agglomeration (scale varying between 1:10000 and 1:25000) the Bezier Spline must be applied. Medium scale map (scale 1:50000), appropriate for presentation of single provinces, may be drawn with sufficient precision utilizing Catmull Splines. For maps covering large regions (i.e. whole countries) simplifying graphical outcome of the system, by applying straight lines joining, may be applied. In any case, the decision on which technique to use is to be made by the system administrator, based on particular situation.

5. CONCLUSIONS

As a result of the undertaken work, an application for dynamic, vector noise maps generation was created. Implementing the application as a browser plugin opens to the public a convenient and free access to the information on the acoustic climate of their surroundings. An unique approach to the map generation and visualization resulted in state-of-the-art quality and performance of the application. Moreover, an unmatched customizability and control over the graphical presentation of the noise maps are delivered to the user. Moving map generation process to the user terminal results in a noticeable decrease for the server-side hardware resources and network bandwidth requirements. The capability to visualize other types of data (i.e. temperature, atmospheric pressure) allows utilizing the application in various environmental monitoring or information systems.

6. ACKNOWLEDGEMENTS

This work was partially supported by the Polish Ministry of Science and Education within the project No. R0201001.

7. REFERENCES

[1] 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, (2002)

[2] A. Czyżewski, J. Kotus, B. Kostek. Determining the Noise Impact on Hearing using Psychoacoustical Noise Dosimeter, Archives of Acoustics, vol. 32, No. 2, pp. 203 – 217, 2007.

[3] Kotus J., "Multimedia System For Environmental Noise Monitoring," *Proc. EURONOISE 2006*, Tampere, Finland, 2006.

[4] The Harmonoise project, access 20.06.2008, http://www.harmonoise.org/

[5] TASK Academic Computer Center, access 20.06.2008, http://www.task.gda.pl

[6] Kozielecki P., Czyzewski A., "A novel noise maps visualization tool," *Proc. 1st IEEE Int. Conf. on Information Technology 2008*, Gdansk, Poland, 2008.

[7] Uklad 2000 – National cartographic projection for Poland, access 20.06.2008, http://www.gis.lublin.pl/index2.php?option=com_content&do_pdf=1&id=12

[8] Snyder J. P., *Map Projections – A Working Manual*, Taylor & Francis Ltd, 1999.

[9] Chang F., Chun-Jen C., Chi-Jen L., "A Linear-Time Component-Labeling Algorithm Using Contour Tracing Technique," Institute of Information Science, Academia Sinica, Taipei, Taiwan.

[10] SWF file format specification, access 20.06.2008, http://www.adobe.com/devnet/swf/pdf/swf_file_format_spec_v9.pdf