RIGA TECHNICAL UNIVERSITY
Faculty of Computer Science and Information Technology
Institute of Applied Computer Systems

Vitaly ZABINIAKO
Doctorate of doctoral program "Computer systems" studies

ANALYSIS AND DEVELOPMENT OF VISUALIZATION METHODS OF GRAPHS IN THREE-DIMENSIONAL SPACE

Summary of doctoral thesis

Scientific supervisor
Dr.sc.ing., associated professor
P.RUSAKOV

Riga 2012

UDK 519.17+004.921.021](043.2)
Za 016 a

Zabiniako V. Analysis and Development of Visualization Methods of Graphs in Three-dimensional Space.
Summary of doctoral thesis.-R.:RTU,
2012.-38 p.

Printed in accordance with the decision of the Board of Institute of Applied Computer Systems, Faculty of Computer Science and Information technology, Riga Technical University No 77 of 26 June, 2012.

This work has been supported by the European Social Fund within the project “Support for the implementation of doctoral studies at Riga Technical University”.

ISBN 978-9934-10-377-3

DOCTORAL THESIS
SUBMITTED FOR THE DOCTORAL DEGREE OF ENGINEERING SCIENCE
AT RIGA TECHNICAL UNIVERSITY

The defence of the thesis submitted for doctoral degree of engineering science will take place at an open session on December 12, 2012 in Meza Street 1/3, auditorium 202, Riga Technical University, Faculty of Computer Science and Information Technology.

OFICIAL OPONENTS:

Professor, Dr.habil.sc.ing. Janis Grundspenkis
Riga Technical University, Latvia

Professor, Dr.sc.comp. Karlis Cerans
University of Latvia, Latvia

Professor, Dr.habil.sc.ing. Gintautas Dzemyda
University of Vilnius, Lithuania

APPROVAL

I confirm that I have developed this thesis submitted for the doctoral degree at Riga Technical University. This thesis has not been submitted for the doctoral degree in any other university.

Vitaly Zabiniako …….. (signature)

Date: ……..

The doctoral thesis is written in English and includes introduction, 4 chapters, conclusions, list of literature and 2 appendices. The main text is 145 pages, it contains 121 figures, 16 tables. The bibliography contains 129 references.

Table of Contents

1. Overall description of the thesis

1.1. Importance of the topic

The concept of information and its computer-aided management became the dominant research area in past few decades, especially in IT (Information technology) domain. A major part of developed software solutions for data storing and processing deals with exploration of different sets of objects, their properties and mutual relationships in order to draw out useful conclusions in scope of particular problem domain. The types of data processing range from automatic (in case if the user provides only the input data and treats its processing as "black box" that outputs expected results) to manual (when execution of data processing steps and their order is fully controlled by the user) with semi-automatic methods lying somewhere in between these two polar approaches. In any case except the fully automatic, the user needs dynamic insight into the structure and current state of the data to enable analysis, guiding the processing and making appropriate conclusions.

As in other fields involving human activity, human-computer interaction model heavily depends on quality of visual feedback, cognition and comprehensive processes. That is why according study known as "information visualization" emerged that seeks effective methods for solving problems in such domains as scientific research, data mining, financial data analysis, logistics, law enforcement, and many others to allow users to see, explore, and understand large amounts of information at once [21].

The full set of possible data types, related visualization approaches, algorithms and techniques is certainly too big to cover within a single academic research, that is why the particular problem was considered – visualization of sets of objects and their mutual relations, known as graphs, that is present virtually in all mentioned problem domains and many more, in form of node-link diagrams, flow charts, layouts, network topologies, etc.

The first and most important aspect which is crucial for any graph drawing is the suitable distribution of its elements in space that conforms both to aesthetic criteria (e.g. minimal edge crossing, maximal symmetry) and emphasizes useful characteristics that might be task-dependent (e.g. explicitly depicting layers of hierarchical structure, if "parent-child" type relationships exist within the graph).

A set of algorithms exist that allows for automatic calculation of graph elements positions, based on the provided adjacency list, adjacency matrix or any other suitable data structure that defines topology of particular graph [5], [6], [18], [20]. The main problem of existing algorithms is that each of these performs best on specific graph structure and choosing the wrong combination may produce unwanted visual artifacts that might even hinder solution of user tasks instead of aiding these. Another important problem is computation complexity that is heavily dependent on number of graph vertices (V) and for some approaches calculating the layout can be up to $\Theta(V^3)$ in worst case scenario.

Although the spatial positioning of graph elements has the most notable influence on data representation, the usage of additional visualization techniques is required during rendering for improvement of exploration capabilities. These techniques range from atomic operations such as zooming of the image of graph for concentrating on a set of elements under inspection and similar approaches [29] to complex visualization strategies [27] that

dynamically alter multiple properties of data by changing colors, shapes, transparency or any other visual elements that represent the information.

Modern computer graphics industry supports data visualization techniques by providing a set of universal API (application programming interfaces) such as Direct2D / Direct3D, OpenGL, Java 3D, etc. that can be adapted to any kind of data drawing, including graph structures. That is why the most important factor regarding visualization techniques is mutual compatibility and the correct order in scope of simultaneous application, rather than the choice of particular API and implementation specifics. The same applies to the compatibility between the visual techniques, layout algorithms and graph data topology that must harmonically coexist within each data visualization session in order to produce visual model that drives solution of user's specific tasks that can be both exploration-oriented and reasoning-oriented.

Separate sub-branch of information visualization discipline exists that is called "visual analytics" devoted specifically to sense-making and reasoning based on analysis of images. The main approaches for benefiting the user within this field are attempts to highlight hidden relationships that are otherwise difficult to perceive, to improve recognition of patterns (for example – by distinguishing clusters), and to reduce search efforts by concentrating large amount of data in a small space.

Traditionally, graphs and diagrams are being drawn on 2D (two-dimensional) planes – book pages, blackboards, posters, etc. The modern computer graphics not only presents the opportunity of interactive 2D drawing, but also establishes software and hardware support for 3D (three-dimensional) space visualization. This introduces one spare dimension for placing of graph elements, allowing for more effective space usage. Even although the third dimension ends up being projected back on 2D plane, visual imitation of graph elements placed in the "depth" positively affects user's comprehension by applying z-buffering and making possible rotations in three-dimensional Euclidean space which is more natural for human's eye-brain perception.

Another important problem that is relevant within graph drawing domain is the evaluation of visualization outcome. The only undisputable characteristic here is the performance that can be expressed and measured quantitatively, for example – in terms of FPS (frames per second). The quality of the produced image, on the contrary, is highly subjective and only few common aesthetic criteria exist that allow to judge whether particular combination of graph layout algorithms and visualization techniques, that is applied to certain topology, is better in one case rather than in another [14], [15].

All mentioned concepts that take part in the visualization process are typically integrated within a single software system called GVS (graph visualization system) that provides an interface for the user for operations with data, cognition, comprehension and other collaboration tasks according to his needs. A number of software applications exists for this purpose, ranging from "ad hoc" solutions for particular application fields to very general graph drawing applications, e.g. "Graphviz", "aiSee", "yFiles", etc. Still, it is possible to distinguish separate logical modules that compose GVS architecture and identify a general suitable framework. Investigating this framework further will allow for effective classification of current and future GVS and might even lead to establishing grounds for automatic construction of such systems for which only structured description of desired capabilities, provided at the input, would be enough for generation of appropriate GVS.

1.2. Goal and tasks of the thesis

The goal of the thesis is to enhance graph visualization and analysis capabilities by proposing improved and mutually compatible versions of layout algorithms and visualization techniques that could be integrated and evaluated within appropriate graph visualization system.

In order to achieve this goal, **a set of specific tasks** has been defined:

- to analyze theoretical background and current situation in the domain of visualization of graphs;
- to outline drawbacks and identify potential improvements of existing approaches;
- to improve visualization methods by modifying existing and designing new appropriate algorithms and techniques;
- to implement proposed visualization methods;
- to set up theoretical background for automated creation of GVS;
- to confirm usability of proposed improvements by experimenting on real datasets and evaluating results.

1.3. Object and subject of the research

The object of the research is the information visualization process and its conformance to end-user tasks in scope of graph drawing.

The subject of the research is a set of graph layout algorithms, visualization techniques and interaction approaches that is integrated and evaluated, based on various datasets regarding:

- interdependencies of committed crimes, acquired from the prototype of the Criminal Procedure Information System which is developed and supervised by the Information Centre of Ministry of Interior of the Republic of Latvia;
- infrastructure of domestic railroad network of the Republic of Latvia;
- general model of air traffic control system.

1.4. Methods of the research

The methods of the research imply usage of graph theory, theory of three-dimensional computer graphics, mathematical statistics, cognitive and perceptual psychology.

1.5. Theoretical importance

The theoretical importance of this thesis is as follows:

- integration of existing algorithms within proposed model of partial hybrid algorithm which has improved usability in application with different types of graphs in comparison with separate methods;
- development of visualization techniques for enhanced comprehension of graph topology and discovery of data patterns by clustering;

- development of new approach for automated visual interpolation of useful target values which are associated with graph vertices;
- development of new approach for color-coded interaction with graph elements, based on magnetic-spring model;
- development of generic GVS model and its modules suitable for general-purpose visualization and analysis of graphs;
- development of specification of framework for GVS auto-construction and formalized evaluation.

1.6. Practical value and approbation

The practical value of this thesis is the implementation of mentioned improvements of algorithms and visualization techniques to allow both explorations of general graphs and analysis of specific datasets. A number of experiments were carried out on a set of data acquired from the prototype of The Criminal Procedure Information System and infrastructure of domestic railroad network of the Republic of Latvia for obtaining useful conclusions in according problem domains.

Results of the thesis were also described in scope of 13 international publications:

1. Zabiniako V., Rusakov P. Simultaneous Application of Graphs and Heightmaps for Enhanced Spatial Analysis. In: Saeed, K., Abraham, A. (Eds.) International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM). Machine Intelligence Research Labs, ISSN 2150-7988, Vol. 3, 2011, pp. 522–529 (indexed in INSPEC database).
2. Zabiniako V., Rusakov P. The Definition of Framework for Automated Creation of Graph Visualization Systems. Scientific Proceedings of Riga Technical University, Computer Science, series 5, volume 47, p. 109-115, 2011 (indexed in ArnetMiner database).
3. Zabiniako V. Using Force-Based Graph Layout for Clustering of Relational Data. In: Grundpenkis, J., Kirikova, M., Manolopoulos, Y., Novickis, L. (Eds.) Advances in Databases and Information Systems (Lecture Notes in Computer Science). Germany: Springer-Verlag Berlin Heidelberg, 2010. pp. 193 – 201 (indexed in SpringerLink database).
4. Zabiniako V., Rusakov P. Combined Representation of Data with Graphs and Heightmaps for Visual Analysis and Spatial Decision Support. In: Xiao, Y., Muffoletto, R., Amon, T. (Eds.) Proceedings of the IADIS International Conferences. Computer Graphics, Visualization, Computer Vision and Image Processing 2010. Germany: IADIS Press, 2010. pp. 184 – 192 (indexed in IADIS digital library).
5. Zabiniako V., Rusakov P. Graph Drawing in Lightweight Software: Conception and Implementation. In: Skala, V., Hitzer, E.M. (Eds.) Proceedings of the GraVisMa 2010 Workshop on Computer Graphics, Computer Vision and Mathematics. Czech Republic: Vaclav Skala – Union Agency, 2010. pp. 151 – 154.
6. Zabiniako V., Rusakov P. Supporting Visual Techniques for Graphs Data Analysis in Three-Dimensional Space. In: Butleris, R., Butkiene R. (Eds.) 17th International

Conference on Information and Software Technologies IT 2011 (Research Communications). Lithuania: Kaunas University of Technology, 2011. pp. 75-87.
7. Zabiniako V., Rusakov P. Comparative Analysis of Visualization Aspects in Technologies Direct3D and OpenGL. Scientific Proceedings of Riga Technical University, Computer Science, series 5, volume 26, p. 209-221, 2006.
8. Zabiniako V., Rusakov P. Analysis of Visualization Problems of Graphs and Models of Graphs. Scientific Proceedings of Riga Technical University, Computer Science, series 5, volume 30, p. 138-148, 2007.
9. Zabiniako V., Rusakov P. Development and Implementation of Partial Hybrid Algorithm for Graphs Visualization. Scientific Proceedings of Riga Technical University, Computer Science, series 5, volume 34, p. 192-203, 2008.
10. Zabiniako V., Rusakov P. Definition of General Requirements for Graph Visualization Software. Scientific Proceedings of Riga Technical University, Computer Science, series 5, volume 47, p. 109-115, 2011 (indexed in ArnetMiner database).
11. Yershov A., Zabiniako V., Semenchuk P. Using Concatenated Steganography for Visual Analysis in GIS SOA. The 52nd International Scientific Conference of Riga Technical University, Latvia, Riga. October 13, 2011 (accepted for publishing).
12. Zabiniako V., Gorbik O. Visualization of Graph-Based Structures for Navigation and Tracking of Real-World Objects. International Journal of Emerging Trends in Computing and Information Sciences. E-ISSN 2218-6301 / ISSN 2079-8407, p. 288-294, 2012 (indexed in Directory of Open Access Journals database).
13. Zabiniako V., Visualization of Graph Structures with Magnetic-Spring Model and Color-Coded Interaction. Baltic DB & IS 2012. Tenth International Baltic Conference on Databases and Information Systems. 2012, Vilnius, Lithuania (accepted for publishing).

According results were presented in 11 international scientific conferences:

1. IADIS International Conferences. Computer Graphics, Visualization, Computer Vision and Image Processing 2010, Germany, Freiburg, July 27-29, 2010. Zabiniako V., Rusakov P. Combined Representation of Data with Graphs and Heightmaps for Visual Analysis and Spatial Decision Support.
2. The 51th RTU International Scientific Conference, Latvia, Riga. October 13-15, 2010. Zabiniako V., Rusakov P. The Definition of Framework for Automated Creation of Graph Visualization Systems.
3. GraVisMa 2010 Workshop on Computer Graphics, Computer Vision and Mathematics, Czech Republic, September 7-10, 2010. Zabiniako V., Rusakov P. Graph Drawing in Lightweight Software: Conception and Implementation.
4. 17th International Conference on Information and Software Technologies, IT 2011, Lithuania, April 27-29, 2011. Zabiniako V., Rusakov P. Supporting Visual Techniques for Graphs Data Analysis in Three-Dimensional Space.
5. The 46th Scientific Conference of Riga Technical University, Computer Science, Applied Computer Systems, October, Riga, Latvia, 2005. Zabiniako V., Rusakov P. Comparative Analysis of Visualization Aspects in Technologies Direct3D and OpenGL.

6. The 47th Scientific Conference of Riga Technical University, Computer Science, Applied Computer Systems, October, Riga, Latvia, 2006. Zabiniako V., Rusakov P. Analysis of Visualization Problems of Graphs and Models of Graphs.
7. The 48th Scientific Conference of Riga Technical University, Computer Science, Applied Computer Systems, October, Riga, Latvia, 2007. Zabiniako V., Rusakov P. Development and Implementation of Partial Hybrid Algorithm for Graphs Visualization.
8. The 49th Scientific Conference of Riga Technical University, Computer Science, Applied Computer Systems, October, Riga, Latvia, 2008. Zabiniako V., Rusakov P. Definition of General Requirements for Graph Visualization Software.
9. 13th East European Conference, ADBIS 2009, Riga, Latvia, September 7-10, 2009. Zabiniako V. Using Force-Based Graph Layout for Clustering of Relational Data.
10. The 52-nd RTU International Scientific Conference, Latvia, Riga. October 13, 2011. Yershov A., Zabiniako V., Semenchuk P. Using Concatenated Steganography for Visual Analysis in GIS SOA.
11. Tenth International Baltic Conference on Databases and Information Systems, Lithuania, Vilnius. July 10, 2012. Zabiniako V. Visualization of Graph Structures with Magnetic-Spring Model and Color-Coded Interaction.

Related results were also published in scope of developed study materials in EU funded project “Development of study module for model driven software development within the study program “Computer Systems”” (contract no. 2007/0080/VPD1/ESF/PIAA/06/APK/3.2.3.2./0008/0007).

1.7. The structure of the thesis

The content of the thesis includes introduction, 4 chapters, conclusions, list of literature and 2 appendices.

The introduction states the importance of the chosen problem domain, defines the goal, tasks, object and subject of the thesis. It also lists the used methods of the research and defines theoretical and practical importance of the results and their approbation.

The first chapter “Main concepts of graph visualization” provides the description of elements of visualization theory summarizes most notable types of modern graph layout algorithms, visualization techniques and states appropriate theoretical background.

The second chapter “Usage of graph visualization for solving user tasks” provides examples of visual analysis and reasoning describes involved cognition processes, their relation to the concept of GVS and makes a summary of existing graph drawing software.

The third chapter “Development of visualization methods” describes improvements proposed by the author. New versions of algorithms, techniques and visualization strategies for data are being specified. The proposed architecture of generic GVS is presented along with specification of framework for automated construction of such systems.

The fourth chapter “Application of proposed improvements” describes a set of experiments carried out within a custom GVS made by the author of this thesis on real-world data and summarizes obtained results.

The conclusive part summaries main achieved results in scope of this thesis and provides description for possible future improvements.

2. Summary of chapters of the thesis

2.1. Main concepts of graph visualization

In scope of this chapter primary related concepts, such as classification of visualization types, complexity, abstraction and formal definitions of graphs are being provided together with detailed description of existing graph layout algorithms and visualization techniques.

Four existing classes of layout algorithms are being described – force-based, orthogonal, hierarchical and radial:

1. Force-based approach is among the most widespread approaches for drawing graphs nowadays. The original description of this method was proposed in 1984 by Peter Eades [5]. The concept of electromechanical system was used by representing vertices with charged steel rings, replacing edges with springs and allowing for this system to move into balanced state that is characterized with minimal energy. Further improvements of this method are described, such as Kamada and Kawai algorithm [8], Davidson and Harel algorithm [4], magnetic-spring approach by Koza Sugiyama [17], etc. Initial placement of vertices in random positions and further iterative refinements yields the pattern for emerging of resulting layout which is common for all force-based approaches as in Fig. 2.1.

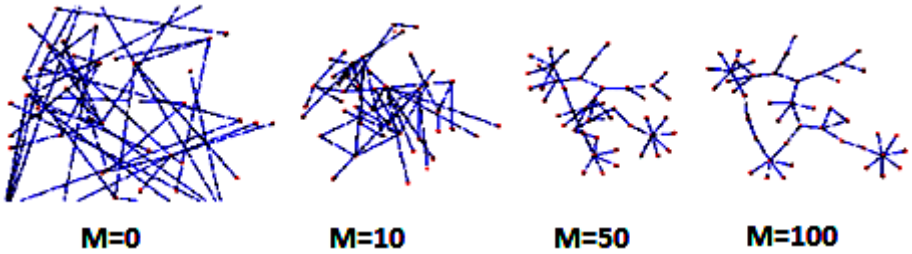

Fig. 2.1. Emerging of force-based graph pattern

2. Orthogonal approach enforces placement of both vertices and edges on a grid with integer coordinates and introduction of $n \cdot (\frac{\pi}{2})$, n $\in$ [1;3] bends for edges. The base framework for obtaining orthogonal layout for graphs is known as "A Topology-Shape-Metrics Approach", originally proposed in [20], [19]. It consists of a chain of three general stages, each of which performs tuning of a particular set of layout properties – planarization, orthogonalization and compaction of the 2D graph model. Application of similar rules in 3D space is also possible, as presented in [12] – this results in layouts as in Fig. 2.2.

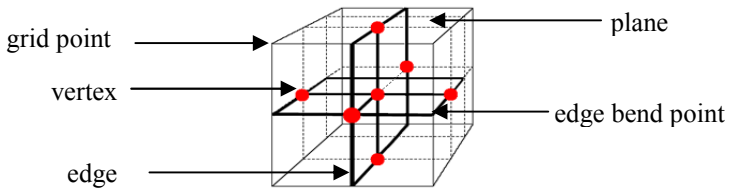

Fig. 2.2. Orthogonal layout in three-dimensional space

3. Hierarchical layout which is suitable for visualization of trees was proposed by [18] which assigns vertices to a set of horizontal layers and seeks for suitable configuration in such aspects as minimal height, width and number of edge crossing, even distribution of vertices within layers and avoidance of long edges. This approach consists of three general stages applied to the initial graph – layer assignment, crossing reduction, coordinate assignment. An extension of this approach to three-dimensional space is presented in [7] which results in 3D layout as in Fig. 2.3.

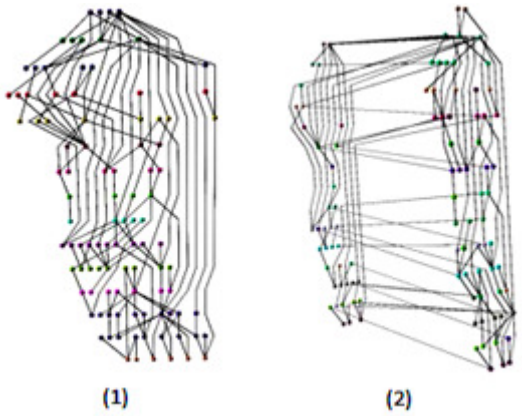

Fig. 2.3. Two-dimensional (1) and three-dimensional (2) graph hierarchical layout

4. In radial-based layout [3], in contrast to hierarchical layout where each layer is represented by a line, the concept of a layer is transferred to a geometrical circle as seen in Fig. 2.4.

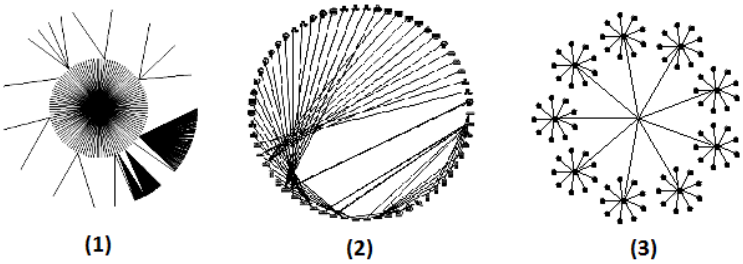

Fig. 2.4. Types of radial-based layout

The authors of [16] propose to visualize hierarchies in 3D by placing a root vertex at the apex of a cone, while the children are uniformly spaced along its base – see Fig. 2.5.

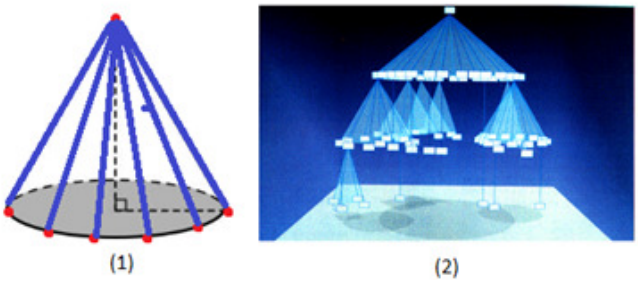

Fig. 2.5. A model for visualizing hierarchies in three dimensions

Visualization of three-dimension objects is a complicated process which foresees an execution of many consequent steps of transformation within the graphic pipeline. Modern visualization systems, such as *Direct3D* or *OpenGL* allow implementing this process while using a set of visualization techniques which concentrate on the outlining useful properties of the graph data without altering its original topology. A set of auxiliary visualization techniques is described in this chapter along with according implementation details, pseudocode, requirements for computer graphics framework and evaluation of its fitness for different tasks (see Fig. 2.6. part 1 for transparency, part 2 for magnification, part 3 for benedictine space, part 4 for blurring/sharpening, part 5 for clustering).

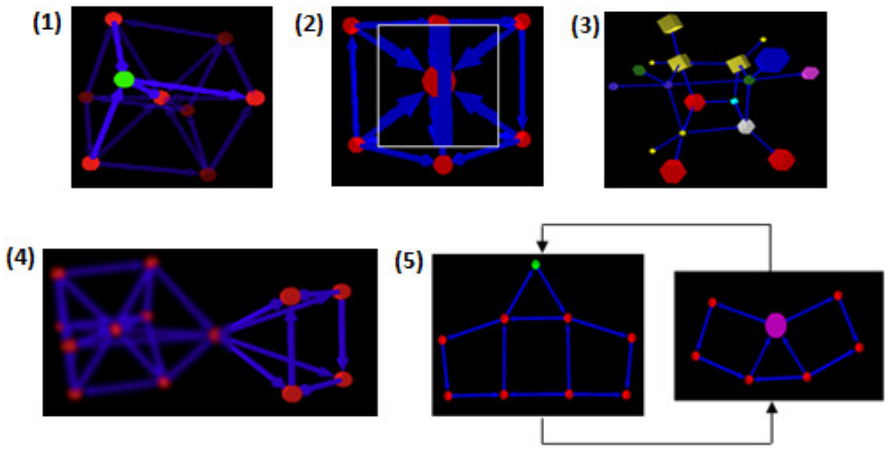

Fig. 2.6. Auxiliary visualization techniques for visualization of graphs

Summary of algorithms and analysis of visualization techniques is presented in the classification at the end of the chapter – refer to Table 2.1.

Table 2.1. Comparison of layout algorithms and visualization techniques

Comparison of layout algorithms				
Category of algorithms	**Advantages**	**Disadvantages**	**Complexity of particular algorithms**	
Force-based layout	Finds the most compact visualization model; allows to present graphs with cycles and is suitable for any general graph.	Requires multiple iterations; a final performance of the particular model is varying and hardly predictable.	Spring model by Peter Eades	$\Theta(\|V\|^2 + \|E\|)$
			Kamada and Kawai algorithm	$\Theta(\|V\|^2 \log\|V\| + \|E\|\|V\|)$ up to $\Theta(\|V\|^3)$
			Davidson-Harrel approach	$\Theta(\|V\|^2\|E\|)$
			Fruchterman–Reingold approach	$\Theta(\|V\| + \|E\|)$
			GEM algorithm	$\Theta(\|V^3\|)$
			Magnetic spring approach	$\Theta(\|V\|^2 + \|E\|)$
Orthogonal layout	A small amount of intersection of edges between layers; the final	A large amount of intersection within edges in	Topology-Shape-Metrics approach	$\Theta((V+c)\min(V,m'+1))$; $\Theta(\|V^2\|)$; $\Theta(\|V^2\|)$

	model is easy perceptible; allows to present graphs with cycles.	layers is possible.	Three-dimensional orthogonal layout	$\Theta(\|V\|)$
Hierarchical layout	Automatic visual distinction of root and leaf vertices; easy comprehendible result and implementation.	A large amount of intersection of edges among layers is possible if graphs have cycles.	Sugiyama approach	$\Theta(\|V\|)$ up to $\Theta(\|V^2\|)$ $\Theta(\|L\|)$ up to $\Theta(\|L^2\|)$
			Three-dimensional hierarchical layout	$\Theta(\|V\|)$ up to $\Theta(\|V^2\|)$ $\Theta(\|V\|)$ $\Theta(\|L\|)$ up to $\Theta(\|L^2\|)$
Radial layout	Automatic visual distinction of root and leaf vertices, zero crossing for visualization of trees.	A large amount of intersection of edges among layers is possible if graphs have cycles.	Radial-based layout	$\Theta(\|V\|)$
			Circular layout	$\Theta((\|V\|+\|E\|)\log(\|V\|))$
			Balloon layout	$\Theta(\|V\|)$
			Cone tree	$\Theta(\|V\|)$

Comparison of visual techniques

Name of the technique	**Desired result**	**Requirements to graphical framework**	**Compatible graph layouts**	**Application domains**
Transparency	Focusing on data under inspection; decreasing influence of other data.	Reading of individual pixel colors from color buffer.	Force-based; hierarchical; orthogonal; radial.	Software requirements analysis; management / tracking of real world objects; Web mining; callgraphs.
Magnification	Focusing on data under inspection.	Matrix manipulation; depth buffer.	Force-based; hierarchical; orthogonal; radial.	General data analysis.
Benedictine space	Representing multiple data attributes in limited space dimensions.	Ability to set size / color / transparency / import custom shapes; depth buffer.	Force-based; hierarchical; orthogonal; radial.	Software requirements analysis; management / tracking of real objects; representation of multidimensional data.
Visual clustering	Finding similar data units and treating them as single object.	Access for selection of vertices via reference pointer; depth buffer.	Force-based; hierarchical.	Software requirements analysis; Web mining; artificial intelligence; social network analysis; bioinformatics, etc.
Data blurring / sharpening	Focusing on data under inspection; dynamically decreasing rate of irrelevant information.	Access to pixel shaders; depth buffer.	Force-based; hierarchical; orthogonal; radial.	General data analysis.

2.2. Usage of graph visualization for solving user tasks

The second chapter provides a description of use cases where visualization of graphs is appropriate and benefits user in particular field, such as model-driven architecture, artificial

intelligence, program visualization systems, the displaying of entity–relationship diagrams, etc.

This chapter describes the process of graph visualization from the interaction point of view between the user and GVS (graph visualization system) – refer to Fig. 2.7.

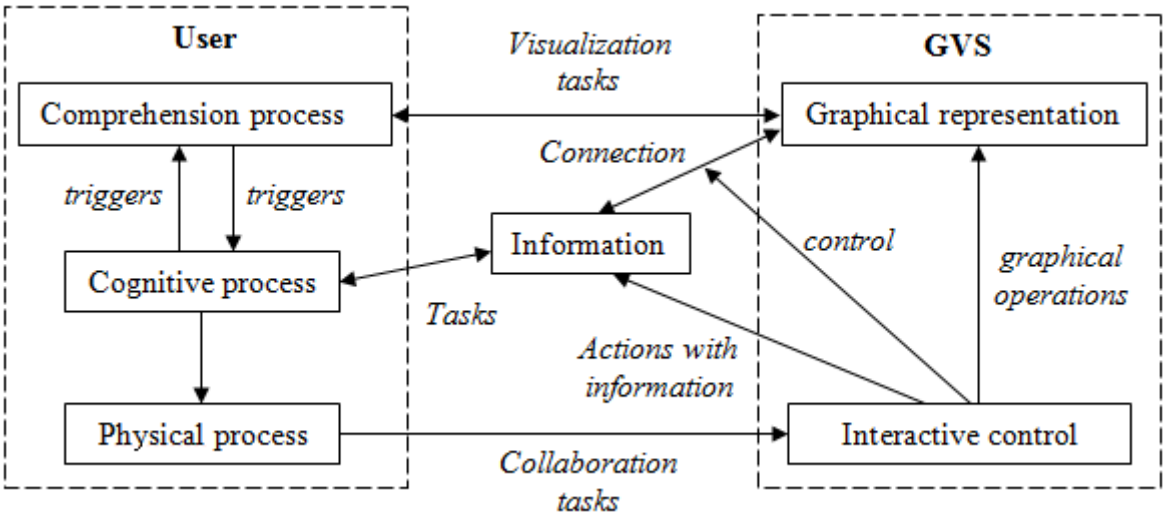

Fig. 2.7. The model of collaboration between user and GVS

The knowledge about properties of the user, the information and necessary tasks of information management allows making a decision on what types of interactive visualization can be used for the information processing. All these factors are important in the process of designing and constructing particular GVS.

This chapter provides a summary of general metrics that allow to judge whether specific resulting graph layout quality is superior to another. Some of the most important of these, according to [1], [14], [15] are as follows – decreasing of mutual crossing and overlapping, decreasing of the occupied space, decreasing of number of edge bends, preserving of coupling of topologically close elements, minimizing total geometric line length, uniform distribution of vertices in a space, positioning vertices with high degree in the center of the drawing, ensuring symmetry of children in hierarchies, maximizing the minimum angles, preserving the single flow of directed edges, etc.

The last part of this chapter is devoted to summarizing destinctive features of existing graph drawing packages, such as Graphviz, Wilmascope 3D, aiSee, Tulip, etc. – see Fig. 2.8.

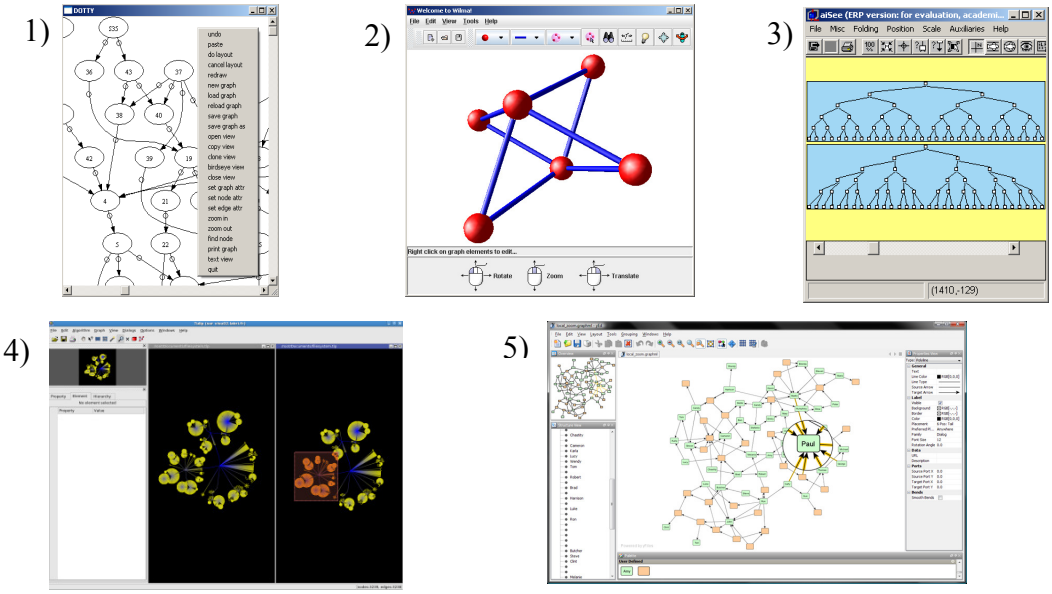

Fig. 2.8. Main windows of existing GVS systems

2.3. Development of visualization methods

The third chapter describes improvements proposed by the author of the thesis in scope of layout algorithms, visualization techniques and additional types of visual analysis.

Previously mentioned algorithms perform visualization tasks successfully if the structure of the graph being visualized is coincident with the particular algorithm class being used (for example, when a hierarchical layout based algorithm processes graph with tree structure which does not contain cycles). Otherwise, the quality of the visualized layout (in terms of such metrics as number of intersected edges, occupied space volume, visual distinguishing of symmetric structures etc.) and its manipulation convenience are on the middle or low level.

The proposed solution is a concept of a hybrid algorithm which allows visualizing information efficiently regardless of the specific features of graph structure. Combination of force-based layout and orthogonal layout (partial hybrid algorithm) is proposed by the author of this work as the first step of creating of full hybrid algorithm. The first layout possesses a useful property to visualize a graph regardless its maximum allowed degree of a vertex, it is also capable to automatically distinguish symmetric structures, although a final performance in this case is hardly predictable due to iterative nature. On the contrary, a performance of the orthogonal layout algorithm is predictable (due to strictly defined steps), although a maximum allowed degree of a vertex in three dimensions in this case is equal to 6.

A development of partial hybrid algorithm by itself is based on concepts of primary and secondary algorithms. After evaluation of both algorithms being integrated, it is possible to conclude, that orthogonal layout algorithms produce bends of edges, which can decrease a comprehension degree. That is why the force-based layout algorithm has chosen as primary, while adding certain concepts of the orthogonal layout algorithm. The pseudocode of the proposed algorithm is shown in Fig. 2.9.

```
/*1*/ for each vertex k
/*2*/    position(k) = point of orthogonal grid(k);
/*3*/ end
/*4*/ until average kinetic energy ( sqrt(dx^2 + dy^2) ) < ξ
/*5*/    for each vertex i
/*6*/       if average kinetic energy > previous average kinetic energy
/*7*/          for each vertex k
/*8*/              position(k) = random offset from position(k);
/*9*/          end
/*10*/      end
/*11*/      previous average kinetic energy = average kinetic energy;
/*12*/      velocity(i) = velocity(i) * f;
/*13*/      for each vertex j
/*14*/         velocity(i) += velocity to repulse from j;
/*15*/      end
/*16*/      for each vertex j connected to i
/*17*/         velocity(i) += velocity to attract to j;
/*18*/      end
/*19*/      position(i) += velocity(i);
/*20*/    end
/*21*/ end
/*22*/ for each vertex k
/*23*/    position(k) = approximation to orthogonal grid(k);
/*24*/    position(k) = position(k) - (min_poz + max_poz)/2;
/*25*/ end
```

Fig. 2.9. Pseudocode of partial hybrid algorithm

Potential improvements in this case are following:

- an initial random scattering of vertices results in unpredictable number of iterations for equilibrium state's search. *Solution*: to place vertices of a graph on closest points of the orthogonal grid before the first iteration will start (this operation is equal to "snap to grid" tool which is commonly used in data editors for decreasing scattering of objects);
- in case if a value of average kinetic energy achieves local minimum, it takes an additional time to retrieve from this. *Solution*: to make a step with random offset from current position, bypassing "slow" iterative retrieving upon detection of local minimum;
- after achieving the equilibrium state, coordinates of vertices holds fractional parts increasing randomness of orientation of model of a graph. *Solution:* forcibly approximate vertices of a graph to points of orthogonal grid after achieving the equilibrium state;
- the equilibrium state is being formed with unpredictable offset from the origin of coordinate system. *Solution*: to place the model of graph near the origin of a coordinate system (similar to orthogonal model), using formula (2.1) for each dimension of vertices of a graph:

$$coord' = coord - \frac{\min\_coord + \max\_coord}{2} \tag{2.1}$$

A set of experiments was performed in chapter 4 to assess the efficiency of the proposed approach.

In scope of this thesis a number of improved techniques for graph visualization are proposed and presented in a similar manner as in chapter 1. According techniques are as follows:

1. Illumination distance – which serves for the focusing on data. The idea is to visually "weaken" graph elements (by reducing color intensity) while being guided with certain pre-set function which takes under inspection geometrical distance between the region of interest and any other arbitrary region. The distance value serves as input to color intensity function, for example – piecewise defined function as in formula (2.2).

$$F(d) = \begin{cases} \frac{1}{d}, d < 10 \\ \frac{1}{10}, 10 \le d \le 20 \\ \frac{1}{d^2}, d > 20 \end{cases} \quad (2.2)$$

2. Gradient stenciling – uses both gradient images and stenciling concept. This technique if useful in case if layout of graph spatial model places vertices in different layers (for example, three-dimensional orthogonal or hierarchical layout). Considering that individual layers might be of particular interest for analysis, it is possible to define general gradient direction in each case (for example "from top to bottom of the screen" or "from left to right"), generate according image and apply it to the color of elements of the graph.
3. Projective shadows – the idea is to draw the three-dimensional data model in iterative steps. The first step captures the model from the current position of the virtual camera like in all previously mentioned techniques. During next steps camera is placed so that it faces model orthogonally – directly from the top, front, left etc. Each rendering result is placed into separate texture. When all steps are complete, textures are placed on corresponding faces of the rectangular parallelepiped (or cube). The parallelepiped is drawn in the scene so that three-dimensional data model is situated at its centre. In this case each texture represents a projection or essentially a "shadow" of original data structure.
4. Auxiliary navigation – in case if information volume that must be perceived or analyzed is big enough, the user must be provided with an informative tool that help him to navigate through the data. This, in turn, requires implementation of "virtual compass". The implementation of this tool is so that the pointer of the compass always points to the pre-defined direction. This enables quick navigation capabilities regardless of current position of virtual camera. Auxiliary navigation might be also adapted to the field of analysis, for example, the "compass" may always point to the parent of selected vertex – this allows improving navigation capabilities between data hierarchies.

According visualization results are presented in Fig. 2.10 (part 1 for illumination distance, part 2 for gradient stenciling, part 3 for projective shadows space, part 4 for auxiliary navigation).

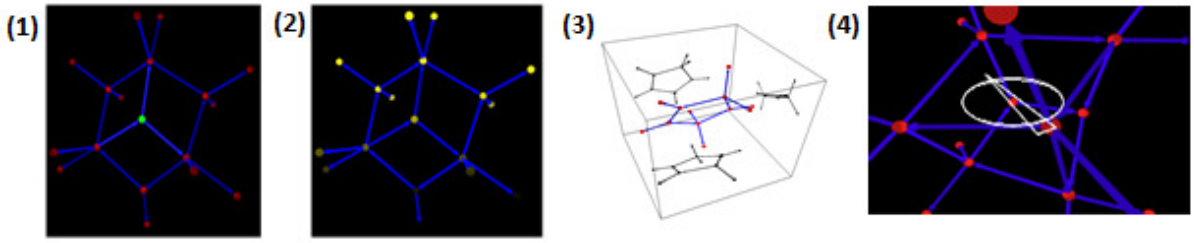

Fig. 2.10. Proposed visualization techniques for visualization of graphs

The summary of the techniques is presented in Table 2.2.

Table 2.2. Summary of proposed visual techniques

Name of the technique	**Desired result**	**Requirements to graphical framework**	**Compatible graph layouts**	**Application domains**
Illumination distance	Focusing on data under inspection; defining custom influence rate of other data.	Ability to define color of object being drawn; depth buffer.	Force-based; hierarchical/tree; orthogonal; radial.	Software requirements analysis; management / tracking of real-world objects.
Gradient stenciling	Focusing on data under inspection; hiding irrelevant information.	Access to stencil buffer.	Hierarchical; orthogonal; radial.	Data with distinctive hierarchical nature.
	Evaluating data complexity; finding best layout for data representation in a plane.	Access for rendering to texture; depth buffer.	Force-based; hierarchical/tree.	Software requirements analysis; management / tracking of real-world objects; tasks where the result must be presented in a plane.
Auxiliary navigation	Maintaining local/global references while navigating through data.	Ability to load visual image of the navigation tool from the file / construct it from built-in primitives.	Force-based; hierarchical/tree; orthogonal.	Tracking of real-world objects; navigation between data hierarchies.

The last part of chapter 3 proposes additional 3D modes for extended analysis – force-based clustering, preliminary visual analysis and color-coded interaction for magnetic-spring graph visualization model.

Clustering of objects is required in case if data units must be evaluated in terms of its similarity or semantic closeness [9]. Examples of such analysis (e.g. in the field of database technology) are optimization of storing data in memory and deriving hidden patterns in large loosely structured datasets.

In the thesis the application of force-based graph layout calculation technique is considered as initial step for clustering of relational data. It is supplemented with custom partitioning strategy in order to provide full-fledged clustering algorithm [24]. The secondary

effect of force-based approach is that the final layout tends to group densely interconnected vertices close to each other, while separating loosely connected groups in different space regions. This is a useful property that highly correlates with clustering logics. While being supplemented with additional recursive space partitioning mechanism it allows to identify regions with separate sets containing unique data elements – see Fig. 2.11.

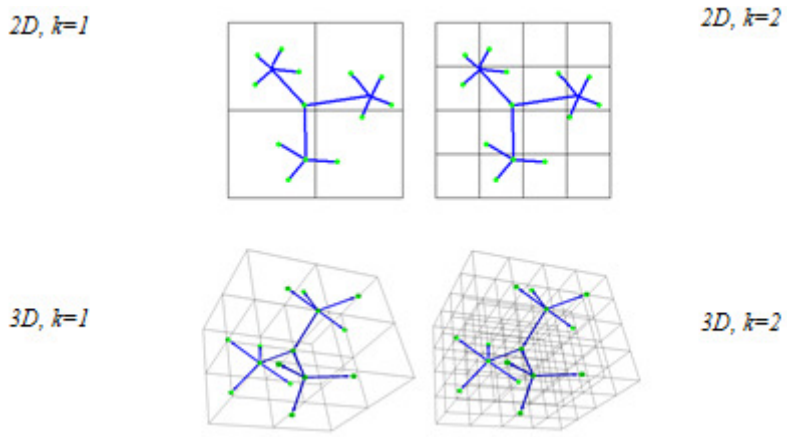

Fig. 2.11. Recursive space partitioning in two and three dimensions

The preliminary visual analysis allows for the user to identify general characteristic and key aspects of the spatial problem under investigation. A good example would be planning of extension of current network infrastructure with adding new elements (stations, hubs, warehouses etc.). At this preliminary stage highly detailed characteristic of the network are pretty much irrelevant because the main emphasis is put on such primary properties as the topology of the network, geographical (taken to the higher abstraction level – geometrical) position of its elements and individual associated quantitative values (throughput \ income \ number of passengers \ transactions, etc.) [22].

Let's consider the following general task: knowing the spatial locations of discrete network vertices, relationships among them and general useful property of each vertex and\or relationship, find according quantitative property for the new network element with desired location. Common sense is averaging \ interpolation \ extrapolation of values of geometrically close network elements. The proposed concept of finding appropriate useful function includes 5 general steps [25]:

1. Visualization of network graph in three-dimensional Cartesian coordinate system using two dimensions for mapping geometric locations of the elements, while using the third dimension for mapping known quantitative values attached to each graph vertex or edge.
2. Drawing this spatial graph into grayscale, considering "weight" of its parts.
3. Extending grayscale graph image to ordinary heightmap.
4. Performing smoothing of the resulting heightmap with two-dimensional image filters.
5. Combining visualization results of steps 1 and 4 for the purposes of visual analysis.

The general schema of data processing flow is presented in Fig. 2.12.

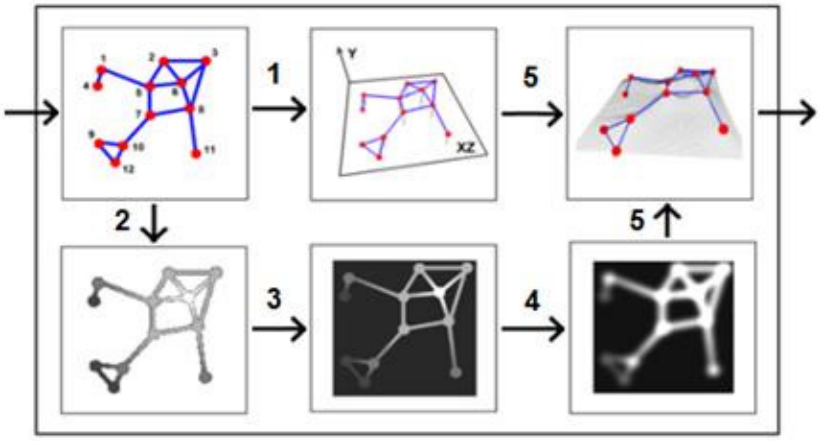

Fig. 2.12. Data processing flow

At the first step the graph body is prepared for visualization in three-dimensional space so that the mapping of the topology occurs in the XZ plane and mapping of the quantitative data is associated with Y axis. The second step involves converting initial graph into grayscale image that will form the basis of the heightmap. During the third step the complete heightmap is formed by filling remaining unoccupied area with black color. It is processed with Gaussian low-pass blur, contrast and brightness to obtain smooth interpolated distribution (fourth step) – see Fig. 2.13.

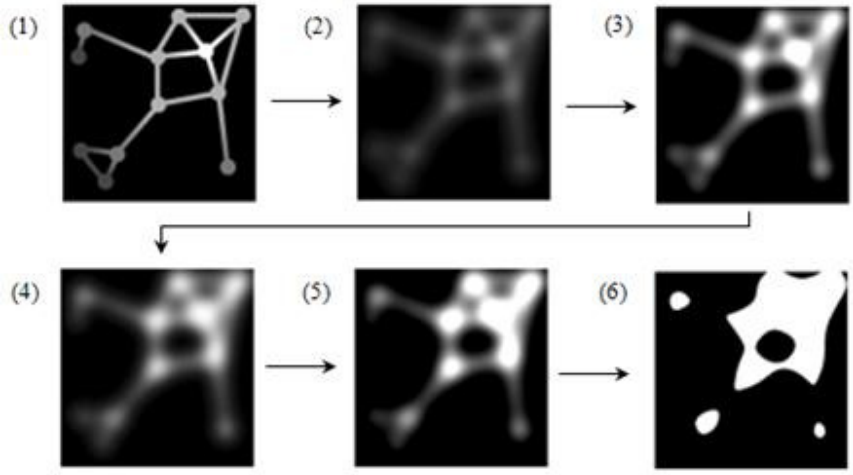

Fig. 2.13. A chain of image processing transformations

During the fifth, final step of the algorithm, the function surface is visualized together with graph body in three dimensions that allows obtaining general combined vision as in Fig. 2.14.

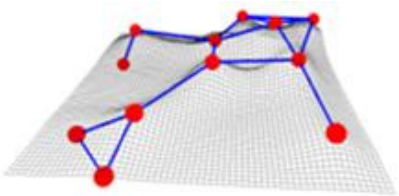

Fig. 2.14. Combined visualization of graph body and function surface

Color-coded interaction for magnetic-spring graph visualization model is based on magnetic-spring variation of force-based approach [23]. This model is enhanced with the color-based encoding strategy which supports dynamic interaction between the user and graph data on a cognitive level. A concept of magnetic force emitter (*MFE*) is introduced which allows for interaction with colored elements of virtual graph model according to a set of rules defined on RGB (Red, Green, Blue) additive color model.

The exploration of properties of force-based layout reveals its potential suitability for visual analysis of the graph which is based on dynamic interaction between the user and GVS. According to the proposed approach, the constant number of iterations (or conditional stop) is removed from the algorithm which leaves the virtual model of a graph in permanent "standby" state during visualization session even when well-formed and balanced layout has emerged. This "semi-active" state of the model enables reactions to additional dynamic force emitters which are positioned and controlled by the user in 3D virtual visualization space.

An example of such interaction would be placing the MFE in chosen space region in order to "attract" to this particular position vertices which conforms to user-defined criteria (e.g. only inherited classes in UML diagram, only non-empty folders in file system structure, etc.) for further inspection. Another possible scenario would be placing the MFE near (or within) the graph body to "repulse" vertices which must be filtered out of the inspection.

Considering that the interaction logic between the MFE and graph elements must be complex enough to allow for composite and selective manipulation with data, while at the same time being intuitively understandable to the user, a set of rules for color-coded cognitive interaction is elaborated.

Author of this thesis proposes two general strategies for color-coded interaction each with two subtypes – "RGB cube color-based proximity", "RGB cube color-based contradistinction", "Hue-based proximity" and "Hue-based contradistinction".

"RGB cube color distance" is the general strategy which relies on the concept of the "distance" D_{RGB} between E and M in the color space of RGB cube. Formally, this metric is defined according to the following formula:

$$D_{RGB} = \frac{\sqrt{(R_M - R_E)^2 + (G_M - G_E)^2 + (B_M - B_E)^2}}{\sqrt{3}} \tag{2.3}$$

A sample Table 2.3 which specifies matrix of D_{RGB} values for primary colors and examples of secondary colors is presented below.

Table 2.3. Set of D_{RGB} values for sample color pairs

(R_E,G_E,B_E) / (R_M,G_M,B_M)	(0.89, 0.51, 0.26)	(0.29, 0.39, 0.85)	(0.18, 0.98, 0.60)	(0.69, 0.20, 0.96)	(0.44, 0.32, 0.20)	(0.13, 0.69, 0.30)	(1.00 1.00, 0.00)	(0.50, 0.50, 0.50)
(0,0,0)	0,61	0,57	0,67	0,69	0,34	0,44	0,82	0,5
(0,0,1)	0,73	0,3	0,62	0,42	0,56	0,58	1	0,5
(0,1,0)	0,6	0,63	0,36	0,83	0,48	0,26	0,58	0,5
(0,1,1)	0,73	0,4	0,25	0,61	0,65	0,35	0,82	0,5

(1,0,0)	0,34	0,68	0,82	0,59	0,39	0,66	0,58	0,5
(1,0,1)	0,52	0,47	0,87	0,21	0,59	0,76	0,82	0,5
(1,1,0)	0,32	0,73	0,59	0,74	0,52	0,56	0	0,5
(1,1,1)	0,52	0,54	0,53	0,5	0,68	0,67	0,58	0,5

Two modifications of this approach are possible. The "proximity" version implies that the strongest magnetic forces (either attraction or repulsion) exist between same or similar-colored pairs of graph elements / magnetic force emitters, while the "contradistinction" version relies on inverted logic and enforces interaction between objects with different (including opposite) colored objects. Within the algorithm according differences are handled via F_m variable which is equal to $(1–D_{RGB})$ in "proximity" version and simply D_{RGB} in "contradistinction" version.

The "Hue-based" strategy is similar to RGB, although it relies on analyzing difference of hue values of interacting elements.

In order to integrate mentioned improvements, a software system "*3DIIVE*" (*Three–Dimensional Interactive Information Visualization Environment*), which includes numerous modules, was built as a part of the research [28]. According software framework on a high abstraction level is presented in Fig. 2.15.

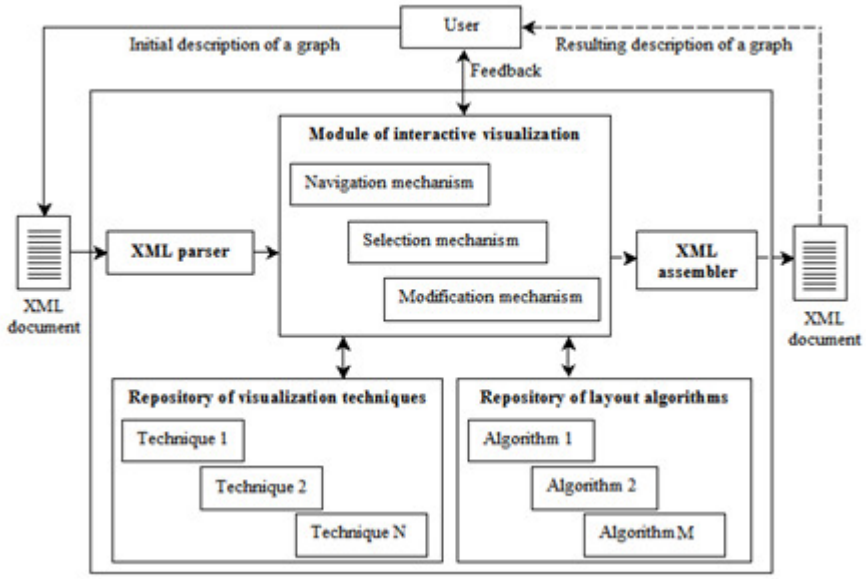

Fig. 2.15. Architecture of software system for graph drawing

The main parts of this system are as follows: *XML parser*, *module of interactive visualization*, *repository of layout algorithms*, *repository of visualization techniques*, and *XML assembler*.

This modular design allows for ability to propose an automated construction of GVS. In scope of this thesis specification of such framework is proposed [26]. It has explicitly defined links between artifacts, dynamic configuration, and formal validation capabilities which should be sufficient for automated generation of GVS. The model of the framework has three-level architecture in which elements within the first level correspond to GVS building

processes, while the second and third levels encapsulate the inner implementation mechanisms – Fig. 2.16.

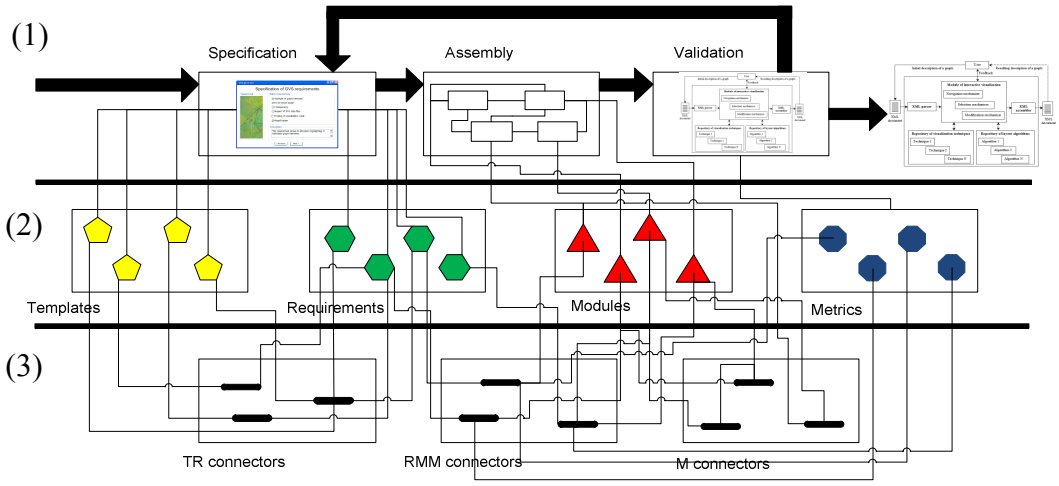

Fig. 2.16. General framework architecture for automated construction of GVS

The top level serves both as entry point for initial requirements specification and the final output in form of appropriate GVS. It consists of three stages – "Specification", "Assembly", and "Validation", that, in general, correspond to classical model "requirements – code – tests". This level also holds the feedback that allows iterative GVS construction, considering the refined user input after preliminary evaluation.

The second level holds a number of sets of individual components that are important parts of GVS (referred as "artifacts" further in text). An artifact is any GVS-related entity that is self-distinguishable and unique. There are four types of artifacts in the proposed framework – "Template", "Requirement", "Module", and "Metric". Each set should be implemented as separate repository with ability to maintain (e.g. add, edit, delete, provide description) the own artifacts and define relationships with other sets via cross-references.

The third level implements the core logic of the framework and holds "connectors" that allow configuring (and reconfiguring) mutual relationships between artifacts, based on a formal mechanism which allow such type of connections between the elements of the framework.

2.4. Application of proposed improvements

A set of experiments was carried out with 3DIIVE GVS in order to assess results of application of proposed improved visualization techniques.

2.4.1. Experiments with the data from the Criminal Procedure Information System prototype

A set of visualization experiments was performed within the prototype of the Criminal Procedure Information System which is an integrated solution of Ministry of Interior of Republic of Latvia which ensures data input, processing and storing in relation to all active criminal cases of Latvia.

The following scenario of visual analysis was performed: two sets of objects were visualized – those which represent criminal proceedings (designated by according numbers) and those which represent persons (designated by abstract unique ID). These objects are visualized as vertices of the graph, while the identified relations between persons and proceeding (i.e. particular person is registered in particular criminal proceeding) are visualized as edges between vertices.

In order to evaluate the performance of proposed partial hybrid algorithm, three sets of data were extracted from the system database:

1) Data registered in the test prototype during the period 01.03.2012 – 31.05.2012 (21 criminal proceedings, 34 persons).
2) Data registered in the test prototype during the period 01.01.2012 – 31.05.2012 (45 criminal proceedings, 63 persons).
3) Data registered in the test prototype during the period 01.01.2011 – 31.05.2012 (259 criminal proceedings, 236 persons).

In all visualization experiments the vertices representing criminal proceeding were colored yellow, vertices representing persons were colored blue, while the edges were colored red.

The visualization was performed using both original force-based and proposed partial hybrid algorithms. Visual result of processing the largest dataset (until a stable equilibrium state was reached) using both algorithms is presented in Fig. 2.17 (part 1 – original force-based algorithm, part 2 – partial hybrid algorithm).

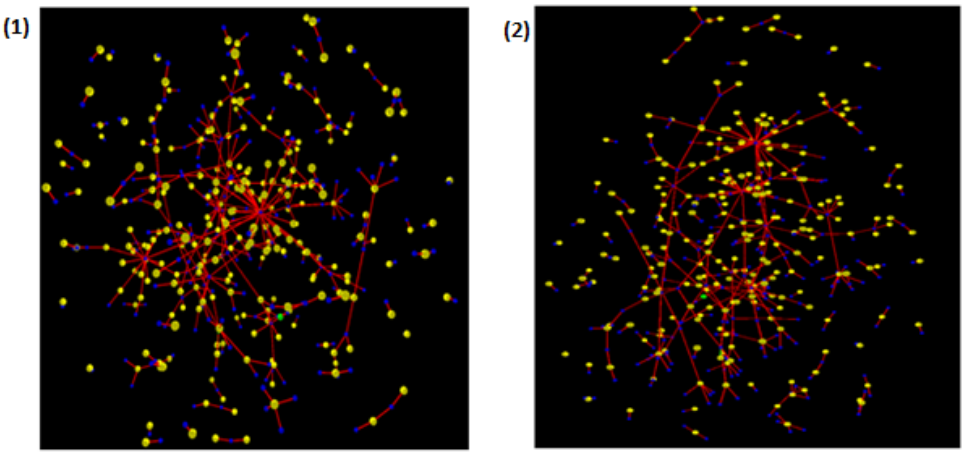

Fig. 2.17. Visual model of a third dataset in stable equilibrium state

Analysis of performance of both algorithms was carried out.

Table 2.4. Quantitative evaluating of performance of algorithms

Force-based layout algorithm							
Dataset 1							
Time (s)	7	8	9	10	11		
Count	6	11	20	9	4		
Frequency	0,12	0,22	0,4	0,18	0,08		
Dataset 2							
Time (s)	16	18	19	20	21	23	24
Count	2	6	11	17	9	3	2
Frequency	0,04	0,12	0,22	0,34	0,18	0,06	0,04

Dataset 3											
Time (s)	48	51	52	56	58	59	60	61	63	68	70
Count	1	2	5	5	8	9	11	4	3	1	1
Frequency	0,02	0,04	0,1	0,1	0,16	0,18	0,22	0,08	0,06	0,02	0,02
Partial hybrid algorithm											
Dataset 1											
Time (s)	4			5			6			7	
Count	2			14			28			6	
Frequency	0,04			0,28			0,56			0,12	
Dataset 2											
Time (s)	14	15		16	17		18	20		21	
Count	3	6		10	20		7	3		1	
Frequency	0,06	0,12		0,2	0,4		0,14	0,06		0,02	
Dataset 3											
Time (s)	49	51	52	56	58	59	60	61	64	65	67
Count	1	2	4	6	9	12	7	4	2	2	1
Frequency	0,02	0,04	0,08	0,12	0,18	0,24	0,14	0,08	0,04	0,04	0,02

The graph of frequency distribution is shown in Fig. 2.18.

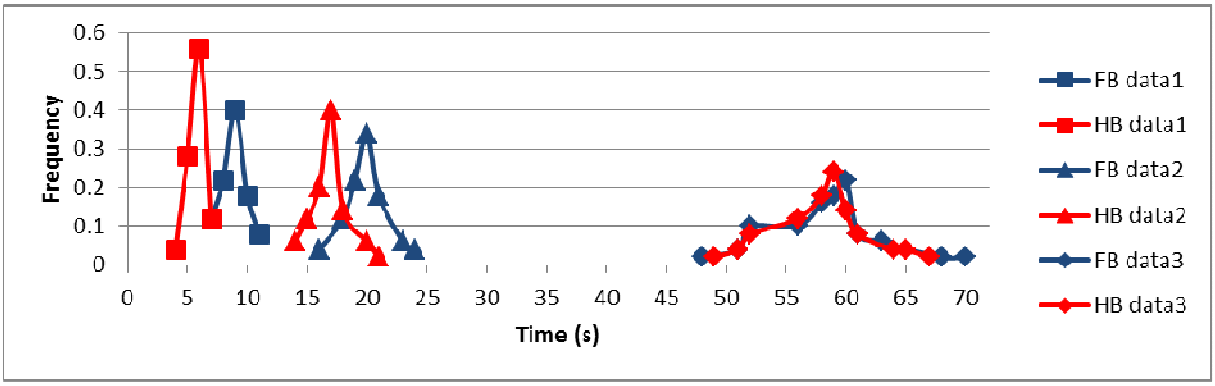

Fig. 2.18. Frequency distribution

The arithmetical mean for FB on dataset 1 is 9 seconds, on dataset 2 is 20.14 seconds, on dataset 3 is 58.72 seconds. Arithmetical mean for HB on dataset 1 is 5.5 seconds, on dataset 2 is 17.28 seconds, on dataset 3 is 58.36 seconds. Analysis of these values reveals that HB clearly outperforms FB on smaller datasets (1–5.5/9)*100=39%; still outperforms it on moderate datasets, although at lesser rate (1–17.28/20.14)*100=14%, and is nearly identical in big datasets (1– 58.36/58.72)*100=1%.

Close-up analysis reveals few distinctive data patterns which are common in all datasets. The first type is “one-to-one”/“one-to-many” type relationship between criminal proceedings and persons. It is visualized as one separate yellow vertex which connects to arbitrary number of blue vertices. The second type is “many-to-many” type relationship which is represented by a chain of interconnected vertices of both colors.

Another visualization scenario was carried out in relation to organizational structure which exists in Ministry of Interior of Republic of Latvia. It demonstrated application of gradient stenciling for putting visual emphasize on current region of interest and hiding irrelevant details of lower levels of the hierarchy – see Fig. 2.19.

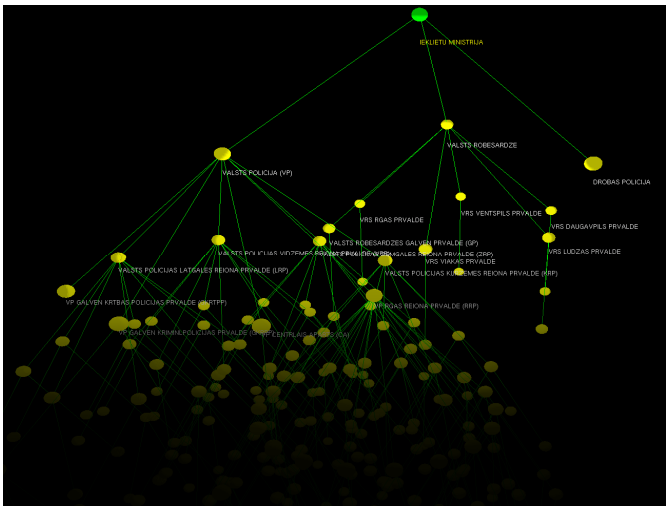

Fig. 2.19. Cone tree with applied gradient stenciling

2.4.2. Application of force-based approach to clustering of relational data

In order to evaluate application of force-based approach to clustering of relational data modified *MS Access* template "*Northwind Traders Sample Database*" [10] was used.

Original database consists of 8 tables, namely: "*Suppliers*", "*Categories*", "*Products*", "*Orders*", "*Order Details*", "*Employees*", "*Customers*" and "*Shippers*". In this example it is enhanced with three additional tables: "*Urgency*", "*Penalty*" and "*Reputation*". Related fields and mutual relationships between tables are shown in Fig. 2.20.

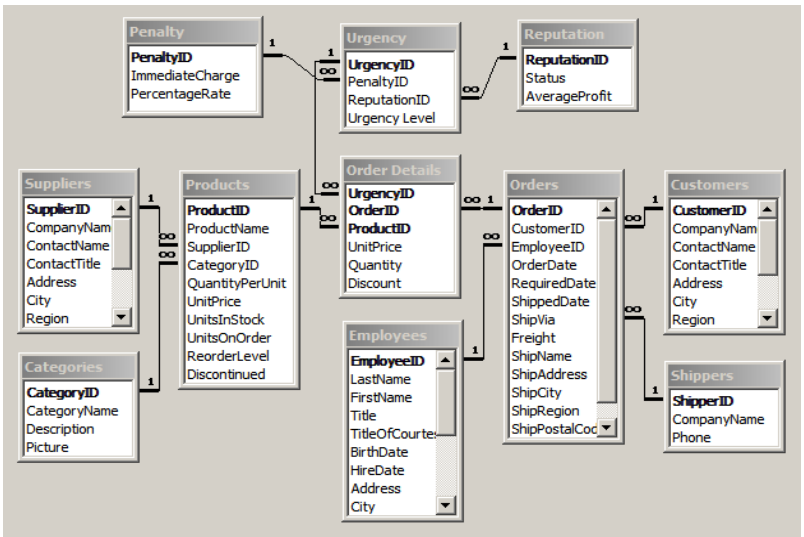

Fig. 2.20. Data relationships in modified "Northwind Traders Sample Database" template

Common visual results of according graph clustering are shown in Fig. 2.21 in 2D (part 1) and 3D (part 2).

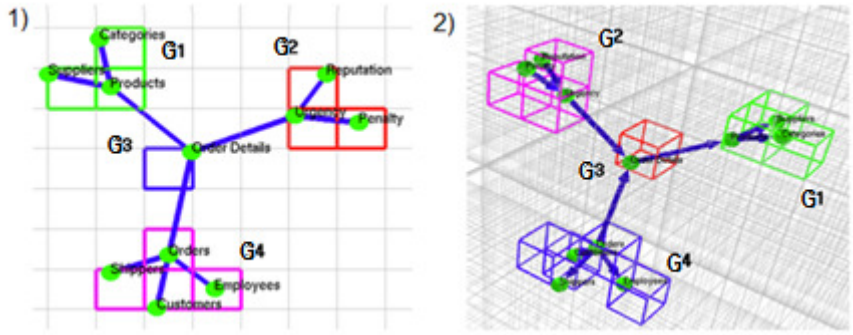

Fig. 2.21. Visual clustering of sample data

In order to visually distinguish different clusters, thicker region borders and individual colors are being used. As it is seen, a set of four clusters (groups) is identified after running implementation of above-mentioned algorithm: G1={"*Suppliers*", "*Categories*", "*Products*"}; G2={"*Reputation*", "*Urgency*", "*Penalty*"}, G3={"*Order Details*"} and G4={"*Orders*", "*Employees*", "*Customers*", "*Shippers*"}. This is a meaningful solution that emphasizes semantic relationships between three separate domains – products, orders and urgency of completion of orders, via fourth logically interconnecting domain – product orders. As it was mentioned above, this information about data groups is useful both for business analysis and data storing optimizations.

Although this certain result is logically valid, it is not the only one possible. Initial random scattering of vertices in a space (before the search of equilibrium state) produces a set of common clustering patterns. In this case these are as follows:

1. The one mentioned above.
2. G1={"*Suppliers*", "*Categories*", "*Products*", "*Order Details*"}; G2={"*Orders*", "*Employees*", "*Customers*", "*Shippers*"}; G3={"*Reputation*", "*Urgency*", "*Penalty*"}.
3. G1={"*Suppliers*", "*Categories*", "*Products*"}; G2={"*Orders*", "*Employees*", "*Customers*", "*Shippers*", "*Order Details*"}; G3={"*Reputation*", "*Urgency*", "*Penalty*"}.
4. G1={"*Suppliers*", "*Categories*", "*Products*"}; G2={"*Orders*", "*Employees*", "*Customers*", "*Shippers*"}; G3={"*Reputation*", "*Urgency*", "*Penalty*", "*Order Details*"}.

As it is seen, boundary domain "*Order Details*" can be trapped inside arbitrary primary group – the one that contains information about orders, products or urgency. These results are also valid, because the border table might be considered as logically related to all three primary domains equally (in case if no additional heuristics was provided by user).

In order to evaluate statistical distribution of these results, a set of experiments was carried out by author to identify frequency of resulting patterns – refer to Table 2.5.

Table 2.5. Distribution of clustering pattern types.

Two-dimensional model	**Pattern type**	**1**	**2**	**3**	**4**
	Count	23	10	6	11
	Frequency	0.46	0.2	0.12	0.22
Three-dimensional model	**Pattern type**	**1**	**2**	**3**	**4**
	Count	26	10	5	9
	Frequency	0.52	0.2	0.1	0.18

The graph of frequency distribution is shown in Fig. 2.22.

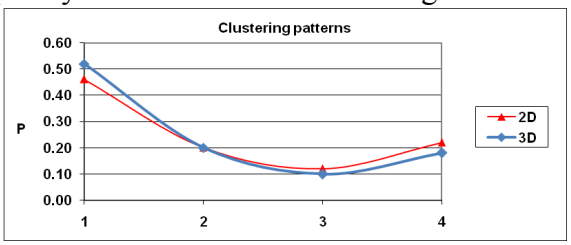

Fig. 2.22. Frequency distribution graph for clustering pattern types

During statistical processing of clustering results, following metrics were acquired: mathematical expectation for two–dimensional model $M=2.1$, statistical dispersion $D=1.45$ and standard deviation $\sigma=1.2$. For the three-dimensional model, these values are following: $M=1.94$, $D=1.36$, $\sigma=1.17$.

2.4.3. Experiments with preliminary visual analysis

In order to briefly evaluate interpolation results produced by the method of preliminary visual analysis the structure of railroad of Latvia was used. Official map of interconnected Latvian cities and routes [13] was chosen as the input – radius of vertices depicts averaged amount of passengers per each station (Fig. 2.23, part 1). This map was converted to appropriate heightmap (Fig. 2.23, part 2) and processed by utilizing according procedure (Fig. 2.23, part 3). The last heightmap visualization step with the distribution of averaged activity is shown – see Fig. 2.23, part 4 for overall perspective, Fig. 2.23, part 5 – for region close up.

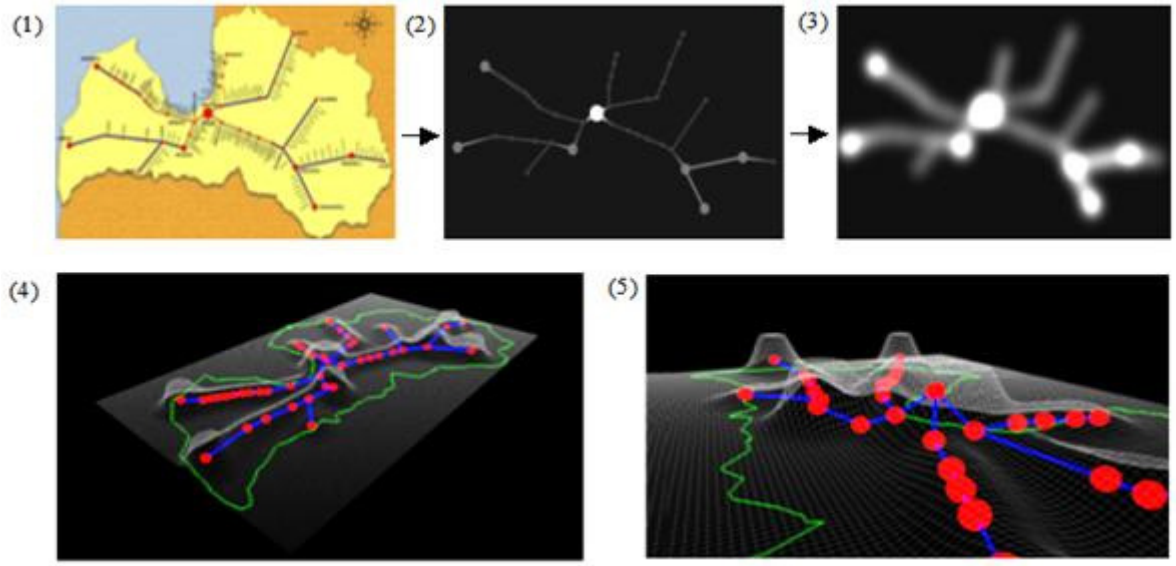

Fig. 2.23. Railroad network of Latvia

In order to locate suitable regions for further expansion, the map of population density in Latvia [2] was used (see Fig. 2.24, part 1).

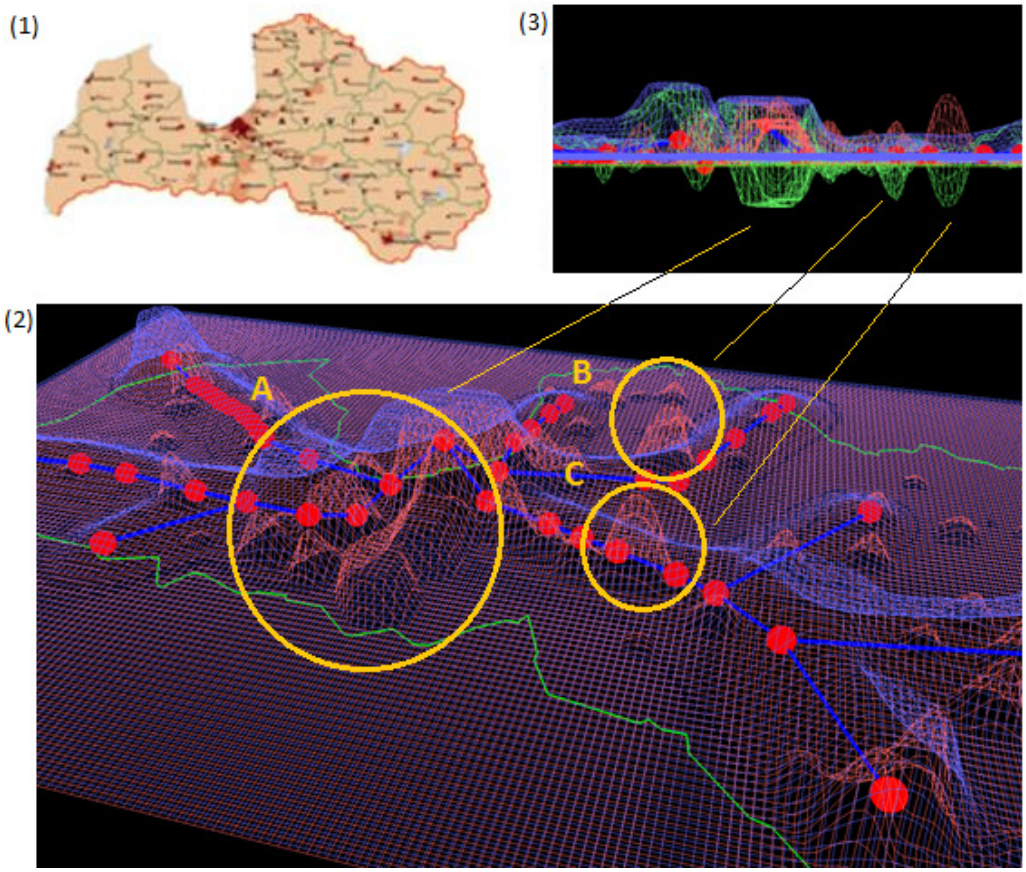

Fig. 2.24. Combination of previous data with averaged population density

According heightmap was extracted from the map of population density and added to the previous result using visual separation with surface color (activity – blue, density – red). In this case color image was desaturated so that only pixel luminance defined the input. As a

result it is possible to visually identify regions with high population level and low railroad coverage (see Fig. 2.24, part 2).

Application of subtraction operator allows identifying on the spot those regions where railroad coverage potential is higher than population density in according area (resulting heightmap values are greater than zero) and those where demand of services might be higher than current infrastructure can offer (resulting heightmap values are less than zero). Visualization of the combined heightmap (green) is presented in Fig. 2.24, part 3 (view along negative Z axis direction in right-handed Cartesian coordinate system).

Preliminary visual analysis of the obtained result reveals that there are several "inconsistency peaks", e.g. region A (district of Bauska – currently no active railroad infrastructure exists), regions B and C (districts of Valmiera and Jekabpils – number of potential passengers is bigger than capacity of the existing infrastructure). Although these images provide useful information about infrastructure of Latvian railroad system and its demographic environment, there was no intention to perform real optimization, as it depends on many other (including historical) relevant aspects and constraints that are not taken into account. As it was stated in theoretical description of the method, the main emphasis is put on preliminary analysis during early planning stage.

2.4.4. Analysis of application of color-based magnetic forces

In order to demonstrate the effect of applying color-based magnetic forces, a case study was carried out with a topological model for library application (refer to [11] for details). Authors of according paper not only defined the appropriate structure of the graph, but also identified the main cycle of system functioning (which is identified by the expert and describes checking out and taking back a book). Fig. 2.25 (part 1) represents the original planar graph. In this example vertices that compose the main cycle are colored yellow, some arbitrary-chosen vertices for MFE interaction demonstration are colored green while the remaining vertices are colored red.

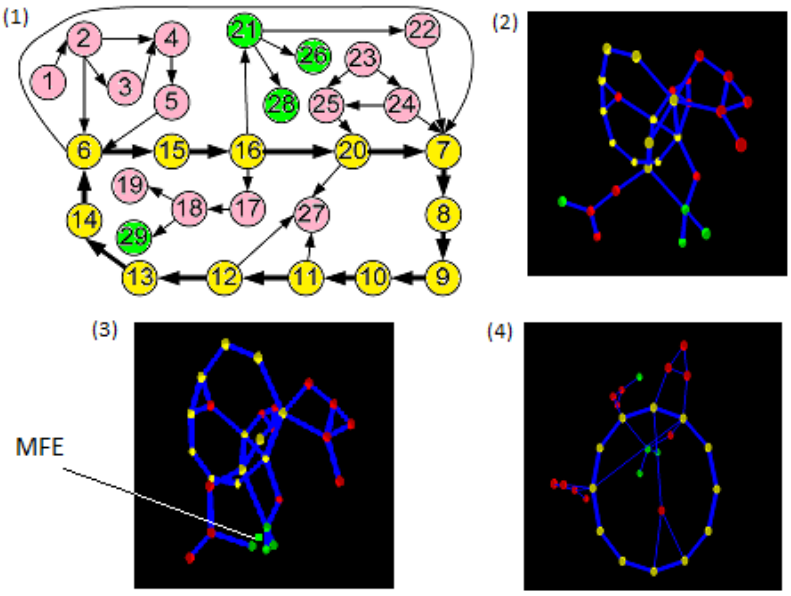

Fig. 2.25. Visualization of topological model for library application

Application of original force-based layout in three-dimensional space yields the initial spatial model (Fig. 2.25, part 2) within the virtual workshop (the "RGB cube color-based proximity" strategy is used in this case study for demonstration purposes).

The next step is the introduction of green-colored MFE attractor which localizes and brings together green-colored vertices for better visual identification. As it can be perceived, movement of those elements that are being affected by MFE, result in appropriate dynamic restructuring of the spatial model (Fig. 2.25, part 3).

Another task which is being performed within this case-study is the distinguishing of the main cycle. For this purpose yellow-colored front magnetic plane is enabled which attracts appropriate vertices (Fig. 2.25, part 4). Additionally, secondary circular layout is applied to all vertices in a front plane, while the width of graph edges on the background is being decreased. This emphasizes visual representation of those vertices and edges that compose the main cycle for better analysis capabilities.

3. Conclusions

Considering that the goal of the thesis was to enhance graph visualization and analysis capabilities by proposing improved and mutually compatible versions of layout algorithms and visualization techniques, this goal is successfully achieved by completing a set of the following tasks:

- *Performing an analysis of theoretical background and current situation in the domain of visualization of graphs*. Historical and modern trends of visualization domain were discussed in this thesis, along with specification of according main concepts (e.g. graph, graph layout, GVS, evaluation metrics, etc.), algorithms, visualization techniques, GVS usage scenarios (taking into consideration related psychological and cognitive aspects) and existing software tools. When appropriate, specifications were supplemented with formal mathematical descriptions and pseudo code.
- *Outlining drawbacks and identifying potential improvements of existing approaches*. Complexity and performance of existing approaches were evaluated together with limitations of existing algorithms in terms of mutual dependencies between structure of the graph being visualized and particular class of the algorithm being used. This allowed identifying potential aspects for further improvement.
- *Improvement of visualization methods by modifying existing and designing new appropriate algorithms and techniques*. A set of visualization methods was proposed which includes partial hybrid algorithm (based on merging useful properties of force-based and orthogonal approaches), improved visualization techniques (illumination distance, gradient stenciling, projective shadows, auxiliary navigation) and additional types of visual analysis (visual clustering, preliminary visual analysis based on combined graphs and heightmaps, color-coded interaction for magnetic-spring visualization approach).
- *Implementation of proposed visualization methods*. Proposed enhanced visualization methods were implemented in scope of 3DIIVE GVS which was constructed by the author of the thesis. This system is capable of visualization of graphs in 3D environment and enables mentioned improvements by applying appropriate algorithms and visualization techniques, as it was demonstrated in chapter 4.
- *Setting up theoretical background for automated creation of GVS*. The specification of three-level architecture was proposed, which is oriented towards automated construction of GVS. Each next level of according framework manages progressively more detailed aspects of construction – that is why all required functionality is covered in the end (starting with high-level definition of requirements for GVS and ending with formal rules of manipulation with separate components that acts as "building bricks" for automated construction of desired GVS).
- *Confirming usability of proposed improvements by experimenting on real datasets and evaluating results*. A set of experiments was carried in scope of originally developed 3DIIVE GVS, which allowed to confirm practical importance of main proposed improvements, based on visualization use cases with the prototype of the

Criminal Procedure Information System, infrastructure of domestic railroad network of the Republic of Latvia, general model of air traffic control system and topological model for library application.

All mentioned improvements and modifications of algorithms and visualization techniques were presented in scope of international scientific conferences and according results were published in international scientific proceedings and journals.

During development of the thesis a number of conclusions were derived both from theoretical studying and practical application experience:

1. Graph layout algorithms and visualization techniques are relatively low-coupled concepts which might be applied in scope of visualization sessions in different combinations or even fully independently (e.g., the graph might be visualized using only partial hybrid algorithm, or this algorithm might be supplied with gradient stenciling, or the gradient stenciling might be used on original data which has no particular spatial pattern). That is why the real task of the visualization scenario is to provide (or to provide an ability to choose) such combination of algorithms and techniques which will benefit solving current tasks of the user.
2. In certain cases, when tasks and analysis needs of the user can be clearly defined, it is possible to predefine specialized approaches which are designed and tuned for this particular task (such as proposed visual clustering for identifying of densely interconnected elements or heightmap-based graph visualization, for convenient spatial interpolation of values associated with graph elements).
3. Still, there is no single united approach for identifying which exactly visual method will benefit solving of arbitrary user task the most, that is why an attempt has been made to propose a concept of more robust hybrid solution which derives and integrates useful properties of existing algorithms and decreases their dependence on compatible graph topologies. Considering that seamless integration of all general classes of existing graph visualization algorithms mentioned in this work is out of scope of single academic research (the complexity of adding each new class tends to be exponential), a specification of full hybrid algorithm was proposed together with implementation of its subclass – partial hybrid algorithm. In scope of this hybrid a combination of force-based and orthogonal approaches was evaluated and confirmed to be suitable for visualization of graphs both from quantitative and qualitative points of view.
4. Another aspect which might influence choosing of particular algorithm or visualization technique is in relation to its implementation specific which also might be affected by external constraints, if such exist. As an example, when the user needs to emphasize certain part of the scene while visually weakening its remaining part, he can choose between illumination distance and gradient stenciling. The final choice is influenced by the fact that the illumination distance is defined with mathematical expressions, that is why in case of complex highlighting patterns construction of appropriate functions is of great difficulty. On the contrary, gradient stenciling provides per-pixel control of the scene but requires stencil buffer, which might be absent on particular visualization hardware (of course, it can be emulated on software level, but this will negatively affect the resulting performance of the visualization).

5. While performing development of new approaches for visualization of graphs, it is possible to seek for opportunities to apply existing general visualization strategies and patterns to this particular problem domain. As an example, usage of projective shadows is well-known technique which is common for all *CAD* systems, but in scope of GVS it allows to evaluate and choose better planar visualization of the graph from several projection alternatives. Similarly, the concept of compass within "auxiliary navigation" is derived from general ability to point on something, but in scope of graph visualization it allows to provide convenient mechanism for dynamic semantic navigation between elements of the graph. At the same time, each particular graph visualization approach being developed may be used in analysis scenarios which deal with semantically different type of input data. As an example, heightmap-based interpolation is useful both in case if the graph represents common 2D coordinates with associated abstract statistic data (as in chapter 4.3) and also in case if it represents full-fledged 3D spatial data (as in chapter 4.4).
6. The author of the thesis presented a description of the architecture of software factory that is capable of generation of GVS systems. The presented framework ensures full automation of gathering of requirements for the system being built and assembling of appropriate software modules. Validation is designed to be partially automated – it is carried out by the user, but the overall rating (conformance to initial requirements expressed) is calculated based on the received input. The quality of GVS being generated depends on the available set of artifacts that is present in the repositories and quality of relationships defined between these. Although there are no particular requirements for the experience of the end user who performs specification and validation of GVS, there are additional requirements for the manager of the framework, who must fully comprehend its architecture. Analysis of existing solutions in this domain revealed that no other semantically close frameworks for visualization had been implemented, although existing general purpose systems utilize similar approach to the automation of software lifecycle.
7. A notable practical achievement is exploring of visual analysis capabilities in scope of real-world problem domain, based on the data of the Criminal Procedure Information System. Chapter 4.1 reveals the possibility to derive useful conclusions which may influence ongoing investigation process by applying visual analysis for identification of potentially related criminal proceedings. This allows for conceptual transition from the current situation when the system is being used as mere data register to the advanced analytical tool which also ensures visual manipulation and comprehension of related objects and entities. The effectiveness of the proposed approach was assessed by a set of quantitative measures which revealed improved performance of visualization up to 39% compared to the existing algorithms.

Literature

[1] F. Beck, M. Burch, S. Diehl. Towards an Aesthetic Dimensions Framework for Dynamic Graph Visualisations. IVIEEE Computer Society. 2009. pp. 592-597.

[2] Best Country Reports, 2007. Population Density Map of Latvia. World Trade Press, Petaluma, USA. / Internet – http://www.bestcountryreports.com/Population_Map_Latvia.php, last visit – February, 2012.

[3] G. Book, N. Keshary. Radial Tree Graph Drawing Algorithm for Representing Large Hierarchies. University of Connecticut. 2001.

[4] R. Davidson, D. Harel. Drawing graphs nicely using simulated annealing. Proceedings of the Symposium on Graph Drawing , 1996. pp. 76-87.

[5] P. Eades. A heuristic for graph drawing. Congressus. Numerantium, Vol. 42, 1984, pp. 149-160.

[6] K. Freivalds, P. Kikusts. Optimum Layout Adjustment Supporting Ordering Constraints in Graph-Like Diagram Drawing. Proc. of the Latvian Academy of Sciences, Section B, Vol. 55 (2001), No. 1, pp. 43-51.

[7] S. Hong, N. Nikolov. Layered Drawings of Directed Graphs in Three Dimensions. Proceedings of the 2005 Asia-Pacific Symposium on Information Visualisation. 2005. pp. 69–74.

[8] T. Kamada, S. Kawai. An algorithm for drawing general undirected graphs. Inf. Process. Lett., Vol. 31, Nr. 1 Amsterdam, The Netherlands, The Netherlands: Elsevier North-Holland, Inc. (1989), pp. 7–15.

[9] M. Kaufmann, D. Wagner. Drawing Graphs: Methods and Models. – Springer. – 2001. – 312 p.

[10] Microsoft Corp. Northwind Traders Sample Database. / Internet – http://www.microsoft.com/downloads, last visit – September, 2011.

[11] J. Osis, U. Doniņš. Formalization of the UML Class Diagrams // Evaluation of Novel Approaches to Software Engineering. Springer-Verlag, 2010. – pp. 180-192.

[12] A. Papakostas, I. Tollis, G. Ioannis. Incremental Orthogonal Graph Drawing in Three Dimensions. Graph Drawing, September 18-20, 1997. pp. 52-63.

[13] Pasažieru Vilciens. Railroad map. Latvijas dzelzeļš, Latvia. 2009. / Internet – http://www.pv.lv/?cat=298, last visit – June, 2012.

[14] H. Purchase. Effective information visualisation: a study of graph drawing aesthetics and algorithms. Interacting with Computers, Vol. 13, Nr. 2. 2000. pp. 147-162.

[15] H. Purchase, S. Bhanji, R. Cohen, M. James. Validating graph drawing aesthetics: A pilot study. Technical Report 336, University of Queensland Department of Computer Science, 1995.

[16] G. Robertson, J. Mackinlay, S. Card. Cone Trees: Animated 3D Visualization Of Hierarchical Information. Proceedings of the SIGCHI conference on Human factors in computing systems New York, NY, USA: ACM. 1991. p. 189--194.

[17] K. Sugiyama, K. Misue: A Simple and Unified Method for Drawing Graphs: Magnetic-Spring Algorithm. Graph Drawing 1994. pp. 364-375.

[18] K. Sugiyama, S. Tagawa, M. Toda. Methods for visual understanding of hierarchical system structures. IEEE Transactions on Systems, Man, and Cybernetics, 11, 1981. pp. 109–125.

[19] R. Tamassia. On embedding a graph in the grid with the minimum number of bends. SIAM J. Comput., vol. 16. no. 3. pp. 421-444, 1987.

[20] R. Tamassia, G. Di Battista, C. Batin. Automatic graph drawing and readability of diagrams. IEEE Transactions on Systems, Man and Cybernetics, SMC-18. pp. 61-79, 1988.

[21] J. Thomas, K. Cook. Illuminating the path: The research and development agenda for visual analytics // IEEE Computer Society – 2005 – 184 p.

[22] A. Yershov, V. Zabiniako, P. Semenchuk. Using Concatenated Steganography for Visual Analysis in GIS SOA. The 52nd International Scientific Conference of Riga Technical University, Latvia, Riga. October 13, 2011 (accepted for publishing).

[23] V. Zabiniako. Visualization of Graph Structures with Magnetic-Spring Model and Color-Coded Interaction. Baltic DB & IS 2012. Tenth International Baltic Conference on Databases and Information Systems. 2012, Vilnius, Lithuania (accepted for publishing).

[24] V. Zabiniako. Using Force-Based Graph Layout for Clustering of Relational Data. In: Grundpenkis, J., Kirikova, M., Manolopoulos, Y., Novickis, L. (Eds.) Advances in Databases and Information Systems (Lecture Notes in Computer Science). Germany: Springer-Verlag Berlin Heidelberg, 2010. pp. 193 – 201.

[25] V. Zabiniako, P. Rusakov. Simultaneous Application of Graphs and Heightmaps for Enhanced Spatial Analysis. In: Saeed, K., Abraham, A. (Eds.) International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM). Machine Intelligence Research Labs, ISSN 2150-7988, Vol. 3, 2011, pp. 522–529.

[26] V. Zabiniako, P. Rusakov. The Definition of Framework for Automated Creation of Graph Visualization Systems. Scientific Proceedings of Riga Technical University, Computer Science, Applied Computer Systems, ser. 5, vol. 47, 2011, pp. 109 – 115.

[27] V. Zabiniako, P. Rusakov. Combined Representation of Data with Graphs and Heightmaps for Visual Analysis and Spatial Decision Support. In: Xiao, Y., Muffoletto, R., Amon, T. (Eds.) Proceedings of the IADIS International Conferences. Computer Graphics, Visualization, Computer Vision and Image Processing 2010. Germany: IADIS Press, 2010. pp. 184 – 192.

[28] V. Zabiniako, P. Rusakov. Graph Drawing in Lightweight Software: Conception and Implementation. In: Skala, V., Hitzer, E.M. (Eds.) Proceedings of the

GraVisMa 2010 Workshop on Computer Graphics, Computer Vision and Mathematics. Czech Republic: Vaclav Skala – Union Agency, 2010. pp. 151 – 154.

[29] V. Zabiniako, P. Rusakov. Supporting Visual Techniques for Graphs Data Analysis in Three-Dimensional Space. In: Butleris, R., Butkiene R. (Eds.) 17th International Conference on Information and Software Technologies IT 2011 (Research Communications). Lithuania: Kaunas University of Technology, 2011. pp. 75-87.