Sławomir WIAK[1], Piotr NAPIERALSKI[2]

Technical University of Lodz, Institute of Mechatronics and Information Systems (1), Institute of Information Technology (2)

Visualization method of magnetic fields with dynamic particle systems

***Abstract**. In this paper we present an interactive method for visualizing three-dimensional vector fields. We decided to apply dynamic particle system, which is directly connected with the values of magnetic field. We introduce a GPU implementation for the simulation stage and rendering stage of a dynamically magnetic fields particle system. The geometry of the machine can be represented as a mesh, giving the possibility to explore the distribution of particle fields in real time.*

***Streszczenie.** W niniejszym artykule zaprezentowana została interaktywna metoda wizualizacji trójwymiarowych pól wektorowych. W celu osiągnięcia zamierzonego celu zastosowano dynamiczny system cząstkowy, gdzie cząstki są bezpośrednio związane z wartościami pola magnetycznego. Wykorzystanie procesora graficznego - GPU na etapie symulacji i etapie renderingu pozwoliła na możliwość zbadania rozkładu pola cząstek w czasie rzeczywistym. (**Wizualizacja pola magnetycznego przy użyciu dynamicznych systemów cząstkowych**)*

Keywords: interactive visualization, vector fields, particle system
Słowa kluczowe: wizualizacja interaktywna, pola wektorowe, system cząsteczkowy

Introduction

The methods for the electromagnetic field presentation using colored maps, isolines and vectors have been developed when the calculation was simplified to two-dimensions. An adaptation of this method to three dimensional spaces is not so simple. One of the main problems is to include the possibility of the depth perceptions of the object situated close at hand. We must remember that the magnetic field is naturally invisible. The visualization of the field density by the colors impedes the perception of geometry of tested devices. The main goal of our research was the creation of effective visualization methods for field calculations results in a virtual reality environment. The methods should consider, better than traditional visualization systems, the spatial and vector character of visualized data.

The basic values describing the magnetic and electric field have vector features that mean direction, turn and magnitude. The knowledge of all these parameters is necessary in most cases to understand the processes performed in an examined system. For example, if the data describe the magnetic field in the three-limb three-phase transformer core then in spite of flux density value also its vector’s direction and turn is the information of major importance for a scientist or an engineer. Only when the spatial distribution of flux density vector is well known the existence of some unfavorable effects can be foreseen (for instance eddy currents) and the methods providing their limitation or elimination can be developed. Taking into consideration that the main goal of scientific data visualization system is the assistance to the researcher in data analysis, the following criteria are assumed:

1) the quality and pictorial character of presentation,
2) the quality of information passing simultaneously,
3) the possibility to observe and to analyse the phenomenon inside the examined volume,
4) the spatial image providing the user with an adequately strong impression of depth and visual comfort,
5) easy orientation in the tested geometry,
6) the exploration of visualized data volume using navigation and interaction,
7) visualization in real time,
8) taking into consideration the movement

The scientific visualization community has concentrated a significant research effort on the design of visualization methods that convey local and global structures occurring at various spatial and temporal scales in flow simulations. In particular, emphasis has been put on the interactivity of the corresponding visual analysis, identified as a critical aspect for the effectiveness of the proposed algorithms.

There are several approaches used for data visualization. The basic classification of existing methods is based on a different method of presenting simulated phenomena. A natural approach seems to be to use a model based on geometry. The geometry of three-dimensional objects can be represented in several ways:

a) Wire-Frame Representation and Polygonal Representation: the object is built of flat polygons (usually triangles or quadrangles), which have common vertices and edges. The grid is defined by the vertices and connections between them. This is the quickest way to create geometry. Processing as defined grid is supported by most graphics cards. Accelerators in the cards are designed to accelerate the operation on the polygon thereby offloading the computer's processor.

b) Volumetric Representation (Voxels): the object is built from elementary cubes (three dimensional pixels). This kind of representation is particularly common in medical diagnostics, where are obtained a number of sections (bitmap images), the patient's body and on this basis creates three-dimensional models. [1,2]

c) Mathematical description - the objects are defined by equations. This can be e.g. bullets, planes, and particularly useful and widely used parametric surfaces (surface patches), for example, Bézier surfaces, NURBS or Hermite.

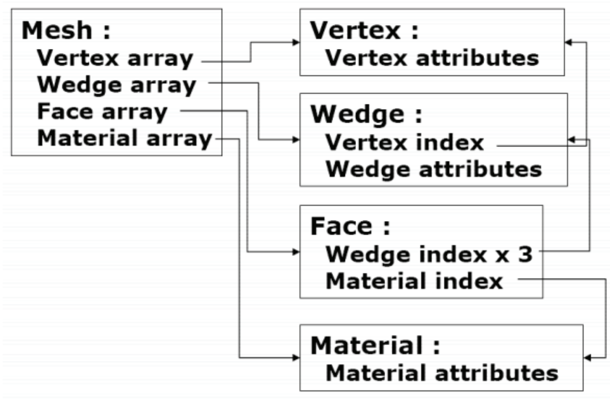

Fig.1. The data structure representation of mesh

The interpretation of a large amount of data is an important part in the design process, and understanding physical phenomena occurring in the devices. Current and most popular methods for visualization of magnetic fields use simple rules of map data to visual attributes, for example by color mapping or volume rendering. Geometry-based techniques generate three-dimensional objects that correspond directly to attributes of the data [3,8,9,15]. One of the challenges in the visualization of magnetic fields is to give the possibility to explore the distribution of particle fields in real time. Feature extraction is an established strategy to provide visual inspection of complex data, while preserving information that is relevant to a given application.

Method description

Particle systems are a technique that allows to create interesting visual effects without much effort [16]. Thanks to them effects can be realized, such as fire, flows, smoke, rain or snow [17,18,19]. Particle systems have been studied in computer graphics for various applications. In this paper, we present sampling and visualizing of three-dimensional unsteady vector fields using scale-space dynamic particles systems. The magnetic field is a vector field. Physical quantities used to describe the magnetic field are the magnetic induction B and magnetic intensity H. A single particle (single macroscopic magnetic dipole) will be characterized by a set of properties that we want to change over time. Particle on the screen will represent the field value at the point of space:

```
class ParticleState{
public:
        Vec3D B;
        Vec3D H;
        float angle;
        float size;
};
 class Particle{
public:
        Vec3D position;
        Vec2D velocity;
        ParticleState current;
        ParticleState change;
        float changeDuration;
        void CalculateChangePerSecond();

private:
        friend class ParticleSystem;

        Particle* nextParticle;
        Particle* prevParticle;
};
```

Fig.2. Particle Class implementation

A particular importance was put on visualization of the field in a way that allows for easy orientation in the geometry of the test device. The geometry of the device is defined in 3D space using mesh of polygons. For this purpose was used a popular file format - obj. The use of this kind of definition allows to import geometry designed in CAD programs. To simulate particles on a GPU [20,21,22,23] we use a „Stateless Particle System", which is defined by a set of start values and the current time (Fig.2). The distribution of starting values for the visualization of the force depends on the value of the magnetic flux. In areas with higher values of the magnetic flux field number of particles of force should be greater than the number of particles in an area with lower value. The particle system stores the velocities, attributes and positions of all particles in textures. These textures are also render targets. Usually in textures is stored image information, but here we used them to store vectors of the field data. In one rendering passing the texture with particle velocities is updated according value of the magnetic flux in current state. The update performs one time step of an iterative integration. Next rendering passing updates the position textures (particles positions) in a similar way, using the loaded data for the position integration. When we use external datas, positions are directly updated from source files. Optionally, positions can be sorted depending on the viewer distance to avoid rendering artifacts and for best visual performance. Positions are transferred from the position texture to a vertex buffer. This geometry data is rendered to the screen in a traditional way as point sprites supported by OpenGL version 1.5 and later. This method renders a perfectly aligned textured 2D square by sending down a single 3D vertex.

Dynamic particle systems on graphics hardware

GPU is a device able to run a large number of task in parallel manner without the time charge associated with processing organization. The simulation on the GPU becomes available in most PC graphics hardware. The calculations are performed by a single computational unit, which are dynamically allocated to the appropriate type of calculation. Recent graphic cards give the ability to define the rendering functionality for Vertex and Fragment shaders.

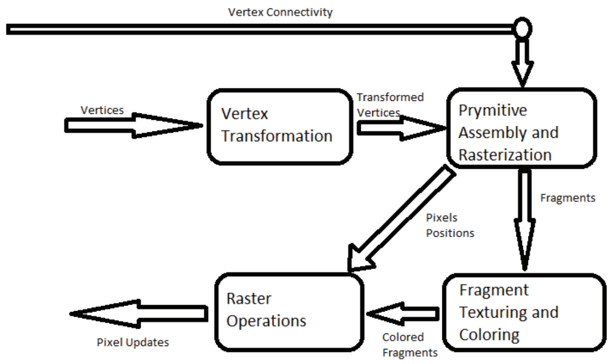

Fig.3. Shader pipeline stages

Vertex shader is run once for each processed vertices. Its task is to transform the vertex position in the virtual 3D space to 2D screen coordinates. It can operate on such properties as the vertex position, color and texture coordinates. Vertex shader output is the input for the next stage in the pipeline, which is either a Fragment shader (Fig. 3). In Fragment shader pixels are taken from the rasterizer, which fills polygons sent from the graphics pipeline. Shaders are used to be administered by the transformation of a large set of items simultaneously, such as on screen, each pixel or for each vertex model. This means parallel processing. The latest multi-core architecture of the GPU is specially adapted to this kind of processing. It is therefore possible to use the GPU to calculate not only the graphics, but also for the calculation of general scientific and engineering solutions. Programmable geometry unit running visualization of magnetic fields with dynamic particle systems is quite efficient. The most important attributes of a particle are its position and value of the magnetic flux. The positions of all particles are stored in a floating point texture with three color components (RGB) that will be treated as x, y and z coordinates. This give a one-dimensional array, texture coordinates representing the array index. The texture is also a render target, so it can be updated with the values and positions of the field. This is not possible to use the same texture as input and output, it is necessary to use pair of current textures with double buffering technique to compute new data from the previous values. A pair of field values textures can be created in the same way as the position textures. Starting points should be dependent on the distribution of magnetic flux. Other particle attributes (like size, color, and opacity) should be defined in such a way that the result gave the best visual effect, so they would need also texture double buffers. This is necessary to store two static values for each particle: its time of birth and a reference to a set of attribute parameters for its particle type from external file. We assume that the particles can be grouped by type of particles in order to minimize static parameters of attributes that must be sent in the final rendering. The geometry of the device is loaded from a separate file and is represented as a mesh of polygons.

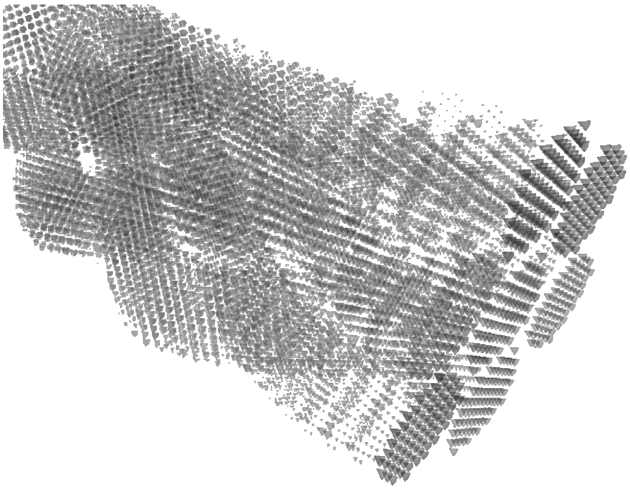

Fig.4. Visualization of magnetic fields with dynamic particle systems

System requires continuous refresh of the attributes of a particles, in same time mesh remains constant. Separating layers for particles and mesh of device allows to ease navigation in the virtual world. The coherent element is the coordinates of mesh and the position of the particles. Field values are represented by the color (RGB) corresponding to a loaded field attributes. An appropriate choice of transparent particles allows clear visualization of the created system.

Results and future works

Calculations were performed for simple visualization of three dimensions. The article examined a model test system with a device surrounded by air. The magnetic field generated by the reference current density J in the device. Forced current flows in a certain direction. We implemented the described algorithm based on OpenGl library as well as the vertex and fragment shaders. In our system, the user shall determine the location of the geometry and the file containing the previously calculated data. The next step is to determine the size of the particles, or choose the automatic settings. Properties of particles can be determined continuously in the user interface. Navigation is made by the mouse and keyboard. Additionally implemented is the ability to display stereoscopic image, that allows to increase feeling depths of viewed scene. Using dynamic particle systems, it is possible to generate illustrations of the magnetic field as discrete particle simulation. The proposed rendering approach permits the rendering of large, time-dependent simulation datasets at interactive rates (36 fps). The presented results show that the animation can be generated using the previously calculated values. The visualization method allows the exploration of virtual space in real time, with the possibility of examining the calculated values. Currently data set are provided to the program from external files. The program uses it to quicken visualization with the possibility of space exploration in real time. Parameters such as particle size and transparency will depend on the position of the observer. Changing these parameters significantly affects the convenience of observation of the phenomena occurring in the device.

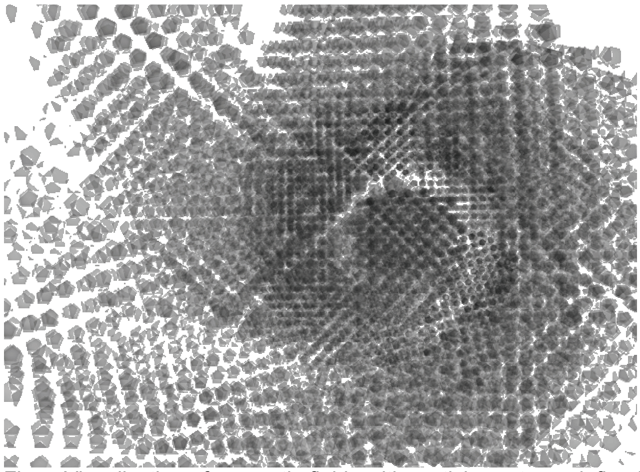

Fig.5. Visualization of magnetic fields with particle systems defined by the user

Using Opengl allows the possibility to transfer applications to other operating systems. The test system was running on the Windows platform. The use of the shaders requires appropriate graphics card. Our work focused on the possibilities of modern graphics cards for fast calculation of field values. The use of several streams may significantly affect not only the aspect of graphics, but it can also be used for complex calculations. Operating on vectors defined by the texture seems to be a reasonable solution to the calculations on large data sets. The results of the visual stage are satisfactory and allow a clear representation of large data sets. Particle system implemented using Opengl proved to be a helpful tool. In the future we will use WebGL - a cross-platform based on OpenGL ES 2.0, exposed through the HTML5 Canvas element as Document Object Model interfaces. This technology allows to implemente our solution directly in the web browser. In order to protect the copyright of created images of the current simulation, special watermarking system [24] will be introduced as well.

REFERENCES

[1] J.P.M. Hultquist, “Interactive Numerical Flow Visualization Using Stream Surfaces,” Computing Systems in Eng., vol. 1, nos. 2-4, pp. 349-353, 1990.

[2] N. Max, B. Becker, and R. Crawfis, “Flow Volumes for Interactive Vector Field Visualization,” Proc. IEEE Visualization Conf. ’93, pp. 19-24, Oct. 1993.

[3] Kruger J., Kipfer P., Kondratieva O., Westermann R. „A Particle System for Interactive Visualization of 3D Flows.“ IEEE Trans. Of Visualization and Computer Graphics., vol 11, no6, 2005, pp 744-756

[4]] Helgeland, A.; Elboth “Hight Quality and Interactive Animations of 3D Time Varying Vector Fields.” IEEE Trans. of Visualization and Computer Graphics., vol 12, issue 6, 2006, pp 1535-1546

[5] D. Weiskopf, G. Erlebacher and T. Ertl, “A Texture-Based Framework for Spacetime-Coherent Visualization of Time-Dependent Vector Fields,” Proc. IEEE Visualization Conf. "03, pp. 107-114, 2003.

[6] Flow Charts: Visualization of Vector Fields on Arbitrary Surfaces, Guo-Shi Li Tricoche, X. Weiskopf, D. Hansen, C.D. Visualization and Computer Graphics, IEEE Transactions on, pp 1067 - 1080 , Volume: 14 Issue: 5, Sept.-Oct. 2008

[7] R.S. Laramee, J.J. van Wijk, B. Jobard and H. Hauser, “ISA and IBFVS: Image Space-Based Visualization of Flow on Surfaces,” IEEE Trans. Visualization and Computer Graphics, vol. 10, no. 6, pp. 637-648, Nov./Dec. 2004.

[8] Generalized Streak Lines: Analysis and Visualization of Boundary Induced Vortices, Wiebel, A. Tricoche, X. Schneider, D. Janicke, H. Scheuermann, G. Visualization and Computer Graphics, IEEE Transactions on On page(s): 1735 - 1742 , Volume: 13 Issue: 6, Nov.-Dec. 2007

[9] Flow Charts: Visualization of Vector Fields on Arbitrary Surfaces, Guo-Shi Li Tricoche, X. Weiskopf, D. Hansen, C.D. Visualization and Computer Graphics, IEEE Transactions on On page(s): 1067 - 1080 , Volume: 14 Issue: 5, Sept.-Oct. 2008

[10] Sampling and Visualizing Creases with Scale-Space Particles, Kindlmann, G.L. Estepar, R.S.J. Smith, S.M. Westin, C.-F. Visualization and Computer Graphics, IEEE Transactions on On page(s): 1415 - 1424 , Volume: 15 Issue: 6, Nov.-Dec. 2009

[11] X. Tricoche, G. Kindlmann and C.-F. Westin, Invariant crease lines for topological and structural analysis of tensor fields. IEEE Trans, on Visualization and Computer Graphics, 14 (6): 1627-1634, 2008

[12] Kindlmann, G.L. Estepar, R.S.J. Smith, S.M. Westin, C.-F Sampling and Visualizing Creases with Scale-Space Particles, Visualization and Computer Graphics, IEEE Transactions onOn page(s): 1415 - 1424 , Volume: 15 Issue: 6, Nov.-Dec. 2009

[13] W Y. Su and J. C. Hart, A programmable particle system framework for shape modeling. In Proc. Shape Modeling and Applications (SMI), pages 114-123, 2005

[14] Helgeland, A. Elboth, T. High-Quality and Interactive Animations of 3D Time-Varying Vector Fields, Visualization and Computer Graphics, IEEE Transactions on pp(s): 1535 - 1546 , Volume: 12 Issue: 6, 2006

[15] Meyer, M.; Whitaker, R.; Kirby, R.M.; Ledergerber, C.; Pfister, H.; Particle-based Sampling and Meshing of Surfaces in multimaterial Volumes. Visualization and Computer Graphics, IEEE Transactions on pp(s): 1539 - 1546 , Volume: 14 Issue: 6, 2008

[16] Batcher, Kenneth E.; Sorting Networks and their Applications. In Spring Joint Computer Conference, AFIPS Proceedings 1968

[17] Buck, Ian; Data Parallel Computing on Graphics Hardware, 2003, Burg2000: van der Burg, John; Building an Advanced Particle System, Game Developer Magazine, 03/2000

[18] Harris, Mark et al.; GPGPU Website, 2003-2004, http://www.gpgpu.org/ Green2003: Green, Simon; Stupid OpenGL Shader Tricks, 2003,

[19] Harris, Mark, Real-Time Cloud Simulation and Rendering, Department of Computer Science, University of North Carolina at Chapel Hill, 2003

[20] A. Wojciechowski, G. Wróblewski, GPU calculated camera collisions detection within a dynamic environment, Springer Lecture Notes in Computer Science 6374, , pp. 357-366, 2010

[21] Jakobsen, Thomas; Advanced Character Physics. In GDC Proceedings 2001

[22] Mark, William R.; Glanville, R. Steven; Akeley, Kurt; Kilgard, Mark J.; Cg: A System for Programming Graphics Hardware in a C-like Language. In SIGGRAPH Proceedings 2003

[23] McAllister, David K.; The Design of an API for Particle Systems, Technical Report, Department of Computer Science, University of North Carolina at Chapel Hill, 2000

[24] Lipiński Piotr, Watermarking software in practical applications P. Bulletin Of The Polish Academy Of Sciences, Technical Sciences Volume: 59 Issue: 1 (2011-01-01) p. 21-25. ISSN: 0239-7528

Authors*: prof. dr hab. inż. Sławomir Wiak, Institute of Mechatronics and Information Systems, Technical University of Lodz, Stefanowskiego 18/22,90-924 Lodz, Poland; dr inż. Piotr Napieralski, Institute of Information Technology, Technical University of Lodz, 90-924 Lodz, Wólczanska 215, Poland E-mail: piotr.napieralski@p.lodz.pl*