A Virtual Reality System for Simulation and Visualization of Discrete Point Charges and its Electric Field

Selan Rodrigues dos Santos†
Antônio Vanderlei dos Santos‡
Luciane Machado Fraga†

URI - Universidade Regional Integrada do Alto Uruguai e das Missões
Departamento de Engenharias e Ciência da Computação†
Departamento de Ciências Exatas e da Terra‡
Caixa Postal 204 – 98.802-470 – Santo Ângelo – RS – Brazil
selan@urisan.tche.br, vando@urisan.tche.br, fraga@urisan.tche.br

Abstract. Recently Scientific Visualization together with Virtual Reality has rapidly become available as leading edge technologies. In the midst of this scenario the demand for educational software that offers support for the teaching process in many areas has greatly increased. This trend is not different in Electromagnetism. This paper presents an educational training system, which is an attempt towards answering this demand. The application has been embedded with both virtual reality technology and scientific visualization techniques in order to carry out interactive investigation and manipulation of concepts concerning electric charge and its electric field. The architectural aspects of such system are discussed jointly with the description of an experimental prototype called Electras.

1. Introduction

Scientific visualization (SV) is an emerging field which has been growing rapidly in the last years. Its main concern is to provide insight into large and complex data sets, depict behavior, and visualize simulation process by way of computer graphics techniques. For this reason SV is considered as a multi-disciplinary field, working as an auxiliary tool which boost up both the research and educational process [McCormick 1987].

Meanwhile Virtual Reality (VR) has become available as a leading edge technology, and it has been equally offering great potential for application in many areas, especially for computer-based simulation. VR is considered as a new and advanced computer interface to 3D models, which brings a completely new world of possibilities concerning human computer interaction. It provides an environment in which users are able to interact and visualize simulations or complex data set, in an interactive fashion. Consequently, VR makes it possible for the user to develop a spatial relationship with the information s/he is interacting with [Brooks 1999].

It is possible to combine this two new technology in order to design high quality simulation software with educational purpose [Barnes 1996]. Such software must provide features like 3D interaction, simulation and visualization of the desired activity. It is important to emphasize that this kind of software is concerned about neither manage nor measuring the learning process itself. Rather, it is designed to provide a mechanism by means of which it is possible to obtain reliable simulation of the process under study.

In this paper we present an architecture for a desktop-based VR simulation system which implements the simulation of electric charges and its field. The system enables users to create, handle and visualize charges, as well as represent electric field through four combining SV techniques. The main goal to be completed by this project is to build a system where the user (usually a student) is able to represent and visualize the

---

† This work is part of a project carried out by the Interdisciplinary Group of Solid State and Simulation at URI.
‡ Sponsored by FAPERSG BIC/99/50585.0
solution of practical problems related to electric fields. The system establishes an
environment in which the user can reproduce book-based problems or even problems
issued by both the teacher and the student itself. For this reason such system can be
categorized as educational software based in scientific visualization techniques and
desktop VR.

The next section introduces the motivation for this work as well as the
description and characterization of the area in Electromagnetism which the software is
aimed at. Section three will discuss the architecture of the system along with its
requirements and operation. Furthermore we will review other related works. In section
four we will highlight the SV techniques that are present in the software in addition to
some VR aspects such as navigation and interaction. The last section consists of the
final considerations and future work.

2. Application

The modern Physics is regarded as important subject especially because of its
application in most areas of human knowledge. However, the misleading understanding
of its concepts and definitions related to both Classical and Modern Physics still is a
problem to be solved.

Concepts like electric charge and electromagnetic wave in Electromagnetism, or
even more complex ones as quantum theory or wave function, are hard to fully
understand and even “visualize”. Indeed, such concepts are proved and verified most of
the time only by mathematical formulas (usually something frightening to students).
Besides, the observer can verify or test great part of those concepts neither by direct
observation nor by experiments carried out in laboratories.

As far as the teaching of Physics is concerned the students usually present a
natural difficult in learning key concepts together with its application in practical
situations. This initial barrier must be overcome for allowing the fulfillment of high
levels of effective and applicable learning.

This natural impediment has had several reasons for its existence like: a weak
mathematical background; most of the time Physics is taught as a theoretical subject
separated from its practical application at real situations in our lives, and; lack of
suitable didactic resources capable of stimulating the students’ curiosity, etc.

As computational technology advances the use of traditional teaching methods
become inefficient and inappropriate. The demand for new effective and modern
solutions lead us to the concept of educational software. The development of a VR
desktop-based system that creates an environment in which the user is able to construct,
visualize, and interact with the simulation of a virtual physically based world could be
regarded as an attempt toward fulfilling this demand.

Such system would play a key role as a complementary tool to the study of
Physics, since it enables the effecting of virtual experiments to the purpose of
enlightening and reinforcing the theoretical understanding in Physics.

In this section we have focused on electromagnetism, giving a general
background as well as the establishment of the limits of the area that the system intends
to solve.

2.1. Electric Charge

The matter found everywhere in our world is made of atoms electrically neutral. The
atom consists of elementary particles bearing an electric charge. Its neutrality comes
from the fact that the positive charge of the nucleus is exactly equal to the total negative
charge of the electrons orbiting the nucleus.

A body is said to be electrified (or charged) when its internal equilibrium
between the positive and negative charges become unbalanced in magnitude as a result
of some process. Nevertheless there is no creation of charge at all, but just the
transferring of charge to another body or even a displacement of charges within the
body. This principle is known as the conservation of electric charge law.
A charged body or elementary particle at rest is surrounded by an electric field called an electrostatic field [Evdokimov 1965]. This elementary particle is the fundamental entity taken into account during the modeling process of our computational system.

The International System of Units (SI) for measuring electric charges is the Coulomb. The fundamental unit of electric charge is \( e \). The proton’s charge is \( e \) whereas the electron’s charge is \(-e\). All charges found in nature are multiples of the fundamental unit \( e \), i.e., they are quantized. Thus, whichever charge \( q \) in nature can be represented by \( q = \pm Ne \), where \( N \) is an integer number. The fundamental unit of the electric charge is given by:

\[
e = 1.60 \times 10^{-19} \text{C} \tag{1}\]

### 2.2. Coulomb’s Law

As mentioned above the electric field surrounds a stationary charged body or particles and acts upon them with a certain force, which is the base for the detection and exploration of such fields. This fact was studied by Charles Coulomb (1736-1806), producing the law that bears his name.

When we place a body with charge \( q_1 \) at a point in the region surrounding another body bearing charge \( q_2 \) we generate a pair of forces \( F \) upon both charges. These forces are directed so that charges with same signal repel one another whereas charges with opposite signal attract each other, as shown in figure 2.1.

![Figure 2-1: Opposite forces upon charges of same (a) and different (b) signal.](image)

According to the principle of superposition\(^4\), force \( F \) may be treated as the result of the interaction of the common electric field with each of the two charges. The Coulomb’s law describes this interaction for point charges\(^5\):

The force exerted in vacuum on each of two point charges by their common electric field is directly proportional to the product of the charges and inversely proportional to the square of the distance between them, i.e.,

\[
F_{12} = \frac{kq_1q_2}{r_{12}^2} \hat{r}_{12} \tag{2}
\]

where \( \hat{r}_{12} \) is the unit vector directed from \( q_2 \) to \( q_1 \) and \( k \) is the Coulomb’s constant which, according to SI, is: \( k = 8.99 \times 10^9 \text{N} \cdot \text{m}^2/\text{C}^2 \). The corresponding force \( F_{21} \) (the force \( q_2 \) exert upon \( q_1 \)) is equal to \(-F_{12}\).

### 2.3. The Electric Field

Initially lets consider a stationary distribution of charges \( q_1, q_2, \ldots, q_n \) in the space. With this in mind lets ignore the forces exerted among them and concentrate only in the effects over another nearby proof charge called \( q_0 \), defined by its Cartesians coordinates \((x, y, z)\). The resultant force at \( q_0 \) is given by:

\[
F_0 = \sum_{j=1}^{n} q_j q_0 \hat{r}_{0jj} \tag{3}
\]

where \( \hat{r}_{0j} \) is the vector parting from the \( j-th \) charge in the system to the point \((x, y, z)\).

---

\(^3\) Defined as the quantity of electric charge that pass through a cross sectional area of a conductor wire during the interval of one second, when inside it circulates an electric current of one ampère.

\(^4\) Whatever electric field is given by the sum of individual electric fields produced by its sources.

\(^5\) A charged body is regarded as a point charge when its linear dimensions are negligible compared with the distance to the points at which the field measurements are made [Evdokimov 1997].
As the force \( \mathbf{F}_0 \) is proportional to \( q_0 \) we can divide it by \( q_0 \), producing a vectorial quantity depending only on the original charge system \( (q_1, q_2, \ldots, q_n) \) and the position \( (x, y, z) \). This vectorial function of \( x, y, z \) is known as electric field \( \mathbf{E} \). The charges \( q_1, q_2, \ldots, q_n \) are called sources of the field. A mathematical definition of \( \mathbf{E} \) is given by the formula [Purcell 1965]:

\[
\mathbf{E}(x, y, z) = \sum_{j=1}^{n} \frac{q_j \mathbf{r}_{0j}}{r_{0j}^2}
\]

(4)

The figure 2.2 illustrates the calculation of \( \mathbf{E} \) for a given set of charges.

![Figure 2-2: Calculation of \( \mathbf{E} \) for a given charge system (q1,q2,q3).](image)

2.4. Electric Field Lines of Force

The visualization of the electric field is usually attained employing two techniques: vector field or lines of force. Calculating and assigning a vector \( \mathbf{E} \) to each point in the space (e.g., in a regular grid) corresponds to the first approach. The second one, more common, consists of the use of lines drawn so that the tangent to them at any point of their length gives the direction of the field strength vector \( \mathbf{E} \) at that point.

The lines of force due to a single point charge are straight radial lines passing through the point charge. They are directed radially outward if the given point charge is positive, and radially inward otherwise, as shown in figure 2-3.

![Figure 2-3: Lines of force for both a positive (a) and a negative (b) charges.](image)

The field strength at any point of the lines of force of an electric field corresponding to a group of several point charges is the vector sum of the field strengths due to each charge taken separately. Figure 2-4 shows two electric fields represented by its lines of force whilst taking into account two point charges.

To the purpose of properly drawing the lines of force it is necessary to follow six elementary rules [Tipler 1996]:

1. The lines of \( \mathbf{E} \) start at the positive charges (or at the infinity) and end at the negative charges (or at the infinity);
2. As the lines of \( \mathbf{E} \) either diverge of converge they are symmetrically disposed around the charge;
3. The number of lines of \( \mathbf{E} \) that either diverge from a positive charge or converge to a negative one is proportional to the charge;
4. The density of lines (i.e. the number of lines per area unit perpendicular to the lines’ direction) around a point is proportional to the value of \( \mathbf{E} \) at this point;

---

6 The SI unit of electric field is Newton/Coulomb (N/C).
5. The lines are radial and uniformly spaced at great distance from the charge system;
6. Whatever two lines cannot have crossing points.

(a)  

(b)  

Figure 2-4: Electric fields corresponding to interaction between two point charges with charge of +6 Coulomb each (a). Opposite charges with +10 Coulomb (red) and −10 Coulomb (green) (b).

2.5. System’s Requirements
Once we have described the fundamental concepts and entities involved in this introductory part of the Electromagnetism, it’s possible to outline the system’s requirements taken into account during the conception of the simulation system.

The major concern that we had borne in mind by the time we started this work was to provide a training system in which the user could interact and build its own experiment in Electromagnetism, acquiring visual insights from it. In addition, the use of such a system in classrooms could serve as a way of executing the assessment task of the system, verifying whether the learning process was satisfactorily accomplished or not.

Equally important was to ensure that the system would grant a reliable and accurate “world”, corresponding exactly to the theorems, laws, and formulas from the books. Jointly, we should offer a mechanism that the user (student) could apply to work out the solution of either problems proposed by the teacher or book-based ones. Thus, it was of primary concern to embed ease interaction features so that the user feels encouraged enough to execute investigation activities over a given experiment.

Together with the above issues it is worth mentioning that VR has lately proved its value when applied to computer-based simulation application. VR potential naturally comes from the fact that it let people visualizing, handling, and interacting with complex data and phenomena. As VR helps user establishes a three-dimensional interaction with the data it has encompassed prototypes, physical models, programming languages, simulation languages and visual interactive simulation [Barnes 1996].

VR is based on three cornerstones: immersion, involvement, and interaction. Its main purpose is to give the illusion of immersion in a computer-generated world. This can be better achieved by means of special equipment that enables users perceive and manifest themselves in another reality through multi-sensorial channels. Nevertheless, there is VR systems that are based only in a desktop peripheral scheme, thus regarded as non-immersive systems [Kirner 1996].

All things considered, our target was to take advantage of VR desktop-based and SV technology in order to develop a computer-generated “artificial” laboratory, in which the user can systematically practice their skills acquired during theoretical classes in electromagnetism. Moreover, such a system intends to be a cost-effective solution in comparison with a real experiment done in experimental laboratory.
3. Architectural Design

Now that we have specified the requirements needed for simulation of electric fields and point charges, we can describe the internal architecture of the Electras (Electric Charge Training System), which stands as our VR simulation system.

There are many issues that a VR educational simulation system such as Electras must address in order to accomplish its objectives. The main factors taken into account during design process were reusing of code for others simulation systems, granting high performance levels in the visualization module, and the design of an easy-to-use interface.

The need for the modularization feature comes from the fact that the project has a long term goal that is the development of a set of educational system to help students visualizing physical phenomena (for example in Mechanics) as well as fundamental mathematical concepts (Analytic Geometry, for instance). Being so, we thought that a straightforward way to reuse pieces of code was utilizing an object-oriented design, obviously in association with an object-oriented language.

To accomplish high performance in the visualization process of the 3D representation of electrical charges and its associated electric field we need a fast graphical API. Despite others similar simulations systems provide a 2-dimensional visual interpretation of electric field, we have decided to take our solution one step further by supporting a 3D view of those concepts. For this reason the system mimics the real behavior of electric charges.

Last but not least concern is to create an accessible interface which permit user to easily interact with, investigate properties of visual entities, and set up most of the situation regarding electric charges demanded by the teacher [El-Khalili 1998].

Therefore, with all these concerns in mind, we have chosen the Visual C++ 5.0 language along with the graphical API OpenGL 1.1 [Kilgard 1997] to implement our prototype.

3.1. Prototype’s Classes

The Electras prototype consists of six main classes corresponding to the entities discussed in the previous section: electric charge (Charge), proof charge (ProofCharge), electric field (EField) which can consist of either lines of force (LineField) or colored field (ColorField) or vector field (VectorField). Also, we have created a class called Environment in which all the instantiated objects will lie in. Figure 3.1 shows how instantiated objects lies inside the Environment object.

Yet in figure 3-1 it is possible to establish the relation of either pertinency or inheritance between some class. Likewise we can see two new classes: Vector and WorldView. Figure 3-2 shows the inheritance relationship between all classes.
Next we will present further information concerning the main classes.

**WorldView Class:** This class is the visual representation of the Environment object created by the application. It comprehends a bounding box representation which offers positioning reference for the elements present in the simulation such as electric charges and proof charges, as suggested in [Keller 1993].

**Environment Class:** It is responsible for the management of the objects created by the user during program execution. It has an internal linked list, which keep track of the objects created by the user. This must be done once we must carry out suitable processing for each object created. The Environment object could be interpreted, in this particular case, as the abstraction of a stationary charge system in a 3D-space. Every time the user inserts a new object in the Environment object it makes sure that this new element will lie inside it by stretching out its limits to adapt to the new situation. Hence, the users are free for creating new elements without caring if the Environment is large enough.

**Entity Class:** This is an abstract class that the following classes are inherited from. This is necessary to grant an independent manipulation of polymorph objects by the Environment class. Thereby, the manager object can process independently all the objects present in the simulation, calling the suitable draw method, which generate the appropriate visualization. This class together with the previous ones are designed to be reused in future applications that demands 3D visualization.

**Charge Class:** This basic class implements the electric charge entity. It has an integer value (positive or negative) expressed in Coulomb that represents the value of an electric charge. The charge is located in a 3D rectangular coordinate system, stored in a Point3D object. The charge could be either hidden or visible inside an Environment object. The charge object is graphically represented by a solid colored sphere, which may assume two colors: red for positive charges, and green for negative ones. Once a charge object is instantiated the user may modify or delete it.

**ProofCharge Class:** This is an extended class form the Charge class. It is quite similar to the Charge class except for the fact that it cannot generate lines of force and its electric value is, by default, +1 Coulomb. These objects are ignored during the calculation of the lines of force; rather it is used just as a depicting mechanism for investigation of specific points in the electric field. More simply, the users may create this object whenever s/he wants to view the force exerted over it by the charge objects present in the system, as shown in figure 2-2.

**LineField Class:** LineField class corresponds to one line of force departing from a positive charge. As mentioned earlier, this is one of the most popular techniques applied in the visualization of electric fields.

**VectorField Class:** The vector field is a 2D regular grid (X, Y, or Z plane). Each point in the grid has an E vector associated to it. The user may set the cell size in the grid.

**ColorField Class:** It is represented by a colored plane corresponding to the equipotentials lines of electric field. Each color is assigned according to a particular threshold extracted from the value of electric field E in a regular 2D grid (X, Y, or Z plane). A simple triangle mesh generates the color plane.
3.2. Other Electric Field Simulation Systems

There are other applications designed to answer the demand of educational systems for Electromagnetism. Most of them are available over the Internet and can be purchased easily. One of the most popular and awarded software is the EM Field 6 [Trowbridge 1998], which is part of the Physics Academic Software project. It covers electric field produced by point charges, line charges, and magnetic field. The electric field can be investigated through electric field vectors, directional arrows, field lines, and electric equipotentials. Although the system presents plenty of capabilities and is fully interactive, it is restrict to a 2D visualization of charge distribution.

Following the same trend we can mention the Electric Field Plotter [Nelson 1996] which has the limitation of allowing the users to set up only nine points charge. Another well-ranked software is the YP Electric Field [Pelletier 1999] which also allows three types of visualization (lines, equipotentials, and electric force vector). However, the main drawback is found in the set up of point charges, which is restrict to a few pre-defined configurations (single, two, triangle and square), bringing obviously constrains to the student’s investigation process.

We can say that the Electras system brings a new approach to such a simulation once it has introduced a true spatial environment in which the user is able to set up point charges in any position inside the 3D bounding box. The RV plays a key role as it improves the investigation task of the simulation by providing to the user the ability of rotating the entire scene, investigating freely the viewpoint that better accomplishes their needs.

Other positive aspect of Electras is that the user can get a volumetric insight about electric field by means of an integrated combination of visualization mechanism. This application is shown in figure 3-3, through the usage of a few equipotential color fields.

Moreover, the 3D environment makes it possible to map the exercises issued in most educational Physics books. These exercises are mainly proposed using 3D geometric shape distribution of point charges like cubes, pyramids, tetrahedron, and boxes. These shapes are impossible to be represented in a typically 2D system.

4. Visualization and RV Techniques

In this section we will give a brief description concerning the usage of Electras system, highlighting the SV and RV techniques that have been applied in it. Figure 4-1 displays the application main window. The system starts showing a bounding box corresponding to an initial world. Although the default units are those from SI, the user is able to change it to CGS unit system by setting World’s properties.
The user can start the intended simulation inserting how many entities as they want like electric charges, proof charges, electric field, and so on. The insertion of a single entity will trigger a property dialog in which entity’s features are set.

After the configuration step the user are able to undertake the exploration phase. For instance, if the user inserts charges then s/he would be able to introduce an electric field and observe the relationship between the charges. Figure 4-2 shows the investigation process using 3D lines of force.

Should the student need to know what is the resulting electric field $E$ in a specific point, s/he could insert a proof charge in that location and check it out by invoking the dialog property upon it (clicking right button mouse), as shown in figure 4-3. However, if the student wants a more general view of the active electric field, s/he can insert the colored field and/or the vector field (figure 4-4).

Vector e Color fields can be created parallel to X, Y, or Z planes in any position in the bounding box. Valuable to mention that the bounding box can be rotated along both X and Y axis, by pressing the arrow keys. Therefore, offering several viewpoints that increase the power of investigation about the simulation.

4.1. Navigation and Feedback

The navigation scheme was design as simple as possible. It is necessary whenever the user performs the insertion of an entity or wants to investigate specific points in space. In this case the user will have to choose amongst two modes proposed in [Foley 1997]: spatial or linguistic. In the spatial positioning task, the user chooses the position according to nearby located objects. In the linguistic approach, the user will specify (in a dialog property window) an exact position $(x,y,z)$ where the object should stay. In both cases the user can get a good guess by looking at the status bar (located at bottom part of the main window) where the current location in 3D space is displayed. Figure 4-5 presents the spatial positioning task.

![Figure 4-5: Spatial positioning task mode.](image)

Two transparent orthogonal planes (XY and XZ) graphically represent the cursor. The crossing of three lines (one is the intersection between both planes, other is a vertical line in XY plane, and the last one is an horizontal line parallel to Z-axis in XZ plane) indicates the current position.

By clicking the right button the user switches between XY navigation (2 DF\(^7\)) or XZ navigation (2 DF) [Lima 1999]. As the user moves the mouse the mentioned lines move proportionally. Thus this approach make available 3 degree of freedom (left, right, up, down, forward, backward) using a mouse device (2 DF). Moreover, the navigation planes were made transparent as a way of prevents the view of being blocked during navigation.

In order to admit the picking-selecting feature we have provided a tree window dialog bar (the small window placed on the left side of the main window) which contains the visual representation of the internal linked list of entities. Therefore, the user may interact or explore all the entities in the system by clicking and selecting the entity’s name on the tree list with the mouse’s left button. By doing so the entity is

---

\(^7\) DF – Degree of freedom
highlighted in the system by means of a small wire-framed bounding box surrounding the graphic representation of the entity (figure 4-3). Clicking over the entity with the right button activates a popup menu in which we can launch a property dialog, hide or delete the entity from the environment.

This approach brings two major advantages. The first one is to avoid dealing with the tricky pick-selection mechanism of the OpenGL API, keeping the project a bit software independent. The second one is the common tree-like representation present in most application (in a Windows Explorer style), what makes the interface more user-friendly.

4.2. Line of Force Drawing Algorithm
The algorithm for plotting the force lines can be divided in two major tasks. The first one consists in calculating the starting points located on the “surface” of the charged particle. In order to allow a 3D distribution we have developed a recursive routine that dispose the lines over a spherical surface in a symmetrically scattered fashion, using spherical coordinates:

```c
1: SphDist(NLines,StartTheta,EndTheta,StartPhi,EndPhi){
2:     if (NLines <= 1) {
3:         SetSphCoords((EndTheta+StartTheta)/2.0,(EndPhi+StartPhi)/2.0);
4:         return;
5:     }
6:     int NLines1stPart = NLines/2;
7:     int NLines2ndPart = NLines - viNLines1stPart;
8:     if ((EndTheta-StartTheta) > (EndPhi-StartPhi)) {
9:         double SliceTheta = (EndTheta-StartTheta)/NLines;
10:        double DivAngle = StartTheta + NLines1stPart*SliceTheta;
11:        SphDist(NLines1stPart,StartTheta,DivAngle,StartPhi,EndPhi);
12:        SphDist(NLines2ndPart,DivAngle,EndTheta,StartPhi,EndPhi,);
13:     } else {
14:         double SlicePhi = (EndPhi-StartPhi)/NLines;
15:        double DivAngle = StartPhi + NLines1stPart*SlicePhi;
16:        SphDist(NLines1stPart,StartTheta,EndTheta,StartPhi,DivAngle);
17:        SphDist(viNLines2ndPart,StartTheta,EndTheta,DivAngle,EndPhi);
18:     }
19: }
20: }
```

The execution of this function is made passing the number of lines to be scattered over the surface and the initial range of the Theta (0 ≤ θ ≤ 2π) and Phi (0 ≤ φ ≤ π) coordinates. The Ro (R) coordinate is the sphere’s ratio, which is the same no matter how many lines must be plotted. The stop condition in line 2 is reached when there is only one line to be set. Hence the position is set in the middle of the spherical region.

In each interaction we split the sphere (or a region of it) in two. Each part receives half of the number of lines (lines 6 and 7). If the Theta range is bigger than the Phi range we divide the Theta range in two, proportionally to the number of lines for each part (lines 9 and 10).

Then we recursively call the function for each halves so it could scatter a smaller number of lines in a smaller region of the original sphere (lines 11 and 12). The same process is undertaken when the Phi range is bigger than the Theta range (lines 15 to 18).

The second major task is to plot the lines departing from the starting points found in the previous step. Basically this object is made of small line segments which direction is obtained from the vector $\mathbf{E}$ calculated at the end point of the previous line segment. The procedure can be summarized as follow:

1. Calculate the electric field of the start point (located in the “surface” of the sphere);
2. Plot a short line to the direction of the electric field;
3. Calculate the electric field at the end of the line;
4. Repeat the steps 2 and 3 until reach another point charge or it has been drawn a pre-defined number of segments.

Although we are talking about a few lines of force per charge, drawing them usually is a time consuming task given that we must take into account the influence exerted by all charges present in the simulation for each small segment. Therefore, executing this calculation demands a special attention to grant high performance, especially because of the considerable number of line segments that has to be calculated and drawn so that the entire line looks like a curved line. These lines could also be visualized in the traditional 2D manner (parallel to X, Y, or Z plane).

Indeed, the more charge objects with force lines the user creates the slower the system will be. Providing that the users is interested in accuracy it would be possible to set line’s property in a way that its line segment would be small enough to grant the smoothness of the lines. However, if the user have decided for speed instead of accuracy it must chose bigger line segments rather than small ones.

5. Conclusions and Future Work

We have briefly described the architectural design for a VR desktop-based simulation system in Electromagnetism. The system has gathered special features from both scientific visualization and virtual reality areas. The key point was to develop a system that could work as an auxiliary tool, supporting the learning process of electrical charges and its electric field concepts by virtual experiments and investigations.

To achieve this goal the system make extensive use of visualization techniques in accordance with [Keller 1993], such as: relating position and scalar value (3D grid plus scalar values represented by labels in the bounding box); description of vector field (lines of force); correlation between vectors and scalars (vector field at the same time with colored field); location of value (use of grid lines helps locating and positioning objects); revealing hidden information (transparency in the navigation mechanism), and; interactive selection and feedback (selection of objects set in 3D space using a mouse device).

The pick-selecting mechanism was implemented by means of a tree-window located in a dialog bar. This approach seems to be valuable once it has avoided dealing with the picking-selecting mechanism of the OpenGL.

The first users have complained about the navigation mechanism. This need to be improved and better represented. It has a lack of depth field and positioning during movement. Furthermore others users have found problems with the correct visualization of the bounding box provoked by the natural dual interpretation performed by their minds, as stated in [Foley 1997].

The following step, besides fixing the mentioned problems up, is to evaluate a beta version of Electras system, considering whether it is actually helping students dealing with basic Electromagnetism concepts or not. Considering Physical aspects, we will extend our system toward the incorporation of the simulation and visualization of the dynamic behavior of the charges. Finally we will have future systems implemented using the Java (plus Java 3D API) platform, as a way of providing collaborative visualization over World Wide Web.

6. References


Figure 4-1: Initial main windows with two charges inserted.

Figure 4-2: Investigating an electric field with 3D lines of force.

Figure 4-3: Investigation of electric field by means of a proof charge.

Figure 4-4: Visualization of electric field through vectors.