Virtual Models of Tesla's Patents

Vladan Vučković[1]

Abstract - **This paper presents theoretical and practical background of the project COMPUTER SIMULATION AND 3-D MODELING OF THE ORIGINAL PATENTS OF NIKOLA TESLA that is applied to the Republic of Serbia's Ministry of Technology and Science by the Museum of Nikola Tesla of Belgrade and the Faculty of Electronic Engineering of Niš. The goal is digitalization and converting of the most interesting and important patents of Nikola Tesla to the digital domain. For the realization of this project, a few software tools are used: AutoDesk 3dsMax and Maya, openGL C++ Libraries and other 3D modeling tools. The part of simulation engine is written in Delphi programming environment.**

Keywords **- 3D Modeling, Computer Simulation, Programing, Rendering Engines.**

I. INTRODUCTION

The computer simulation is very developed and used tool in theory and practice [1],[2]. It is also a very interesting branch of research in general. The project *COMPUTER SIMULATION AND 3-D MODELING OF THE ORIGINAL PATENTS OF NIKOLA TESLA* that is applied to the Republic of Serbia's Ministry of Technology and Science by the Museum of Nikola Tesla of Belgrade and the Faculty of Electronic Engineering of Niš has purpose to digitalize and convert of the most interesting and important patents of Nikola Tesla to the digital domain.

Our team manually convert 3D models using *Autodesk 3dmax* application, based on the original Tesla's patents, that is in Museum's posetion. Then, using those static models, some dynamics are added. We use the most advanced software tools; we render, animate, simulate and visualize the patents. The application that were used are *3ds Max,Adobe Photoshop,RealFlow,V-Ray,Microsoft Visual Studio.*

The computer technology enables that models could be used in unlimited number of copies and in a very different environments. Also, using *openGL, C++, Delphi* and machine language, we generate real time 3D models, rotating and translating using the direct user's commands. The very interesting application is to use stereoscopic vision of the 3D patents.

The main plan of the technological project is to develop models covering 4 main areas of Tesla's work: Tesla's fountain (that integrates machine technology, induction motor, light technology, the physics of fluids), Tesla's electromagnetic motors and rotating field invention, Radio technique and wireless propagation of signals and energy (also with Tesla's coil as the basic invention) and the Tesla's vehicle (ship-robot). To comply this ambitious plan, two teams of professors and students on Electronic faculty are assembled as well as professional experts in Museum.

The first Tesla's patent that was under our attention was Tesla's fountain. We were working almost 6 months of that segment of project and results are very interesting. We use different tools to generate digital model of Tesla's fountain in various configurations and environments. At the end of this convey, we will illustrate some of our results in real 3D rendering models of fountain.

II. TESLA FOUNTAIN PROJECT AND PARTICLE SIMULATION ENGINE

As the result of the development of the new technologies and informatics a novelty approach in modeling and simulation of the Tesla's Fountain was discovered. The idea, which was based naturally by the cooperation between Museum of Nikola Tesla from Belgrade and Faculty of Electronic Engineering from Nis, is to use original Tesla's patents like Fountain to create computer models and simulation using the most advanced software. That was origin of the «Computer Simulation and 3-D modeling of the original patents of Nikola Tesla» technique project TP12210 approved by the Ministry of Science and Technology of the Republic of Serbia. The author is also the project coordinator. As the result of a long period of development, our team has developed Tesla Fountain, based on Tesla's US Patent 1,113,716 Fountain (Fig. 1.). The computer model of Tesla's Fountain contains around 500000 points. This model is a base for rendering and computer simulation/animation.

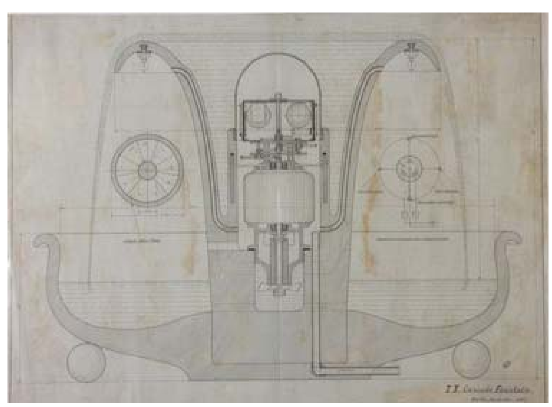

Fig. 1. *Tesla –Tiffany Cascade Fountain*

[1] University of Niš, Faculty of Electronic Engineering, Aleksandra Medvedeva 14, 18000 Niš, (e-mail: vladan.vuckovic@elfak.ni.ac.rs)

Nikola Tesla presented mankind with a huge number of significant inventions, and the ingenuity of these solutions, universality of their application, and their sheer expansion of numerous scientific and technological boundaries, have changed the world around us. The long list totaling 280 patents - 116 original ones - protecting 125 Tesla's inventions, as well as 164 analogues registered in 26 countries of the world, are the testimony of his great and imaginative mind, with their great achievements of equal success in a wide range of fields - from electro and radio technology, to machine engineering, medicine, aviation, and many other areas of science and technology.

„The invention has an unlimited field of use in private dwellings, hotels, theaters, concert halls, hospitals, aquaria, and, particularly, in squares, gardens and parks in which it may be carried out on large scale so as to afford a magnificent spectacle far more captivating and stimulating to the public than the insignificant displays now in use.“

(FROM TESLA'S ORIGINAL PATENT LETTER NO. 1.113.716.)

III. FLUID QUANTIZATION

Using the large amount of quantum's, it is possible to simulate the fluids movement in computer models. The author's application could be used to simulate fluid movements inside the solid structures as well as electrical charged components.

Simulation of particles and fluids in modern graphics is one of the hardest and most demanding fields of work [3]. The main challenge is to make a realistic water or fluid surface. The market demands realistic show, which can be made by a good application with great hardware requirements.

The author's application is foreseen to use with other commercial or non-commercial software.

Real Flow is one such application. RealFlow(RF) is a 32 and 64 bits cross-platform simulation program that runs on Windows and Linux. It consists on a complete set of simulation engines that cooperate to solve complex scenes with multiple interactions among elements of different nature. RF computes object dynamics, fluids, particles, soft bodies, water surfaces... allowing as we said, the interaction of these elements. It also has got some other technological functionality as meshing particles or scripting possibilities.

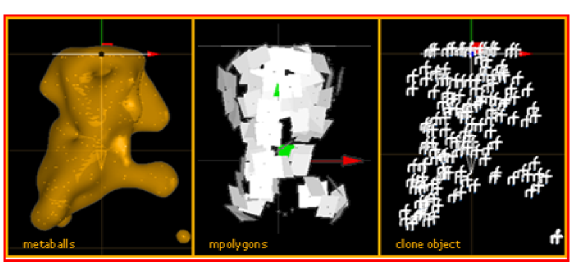

Fig. 2. *Particle structure in RealFlow.*

Fluid particles could be divided into the several categories: gas particles, particles (dumb), fluid surface (real wave) (Fig.2.).

Fluid particles: RF's "liquid" type represents a real incompressible liquid using a collection of coherent particles, which sample the properties of the liquid volume. Each particle represents an element of liquid mass and is also an irregular sample of the real physical fields, which define the liquid's dynamic behavior, such as pressure, velocity, acceleration, density and viscosity. RF's "liquid" particles simultaneously maintain a good sampling of the incompressible liquid volume and provide an accurate representation of the liquid's dynamic properties.

Mesh: RF includes an automatic mesh generator that is able to build polygonal surfaces from particles. There are 3 types of these generators:

- Meatballs: a polygonal cover is build from the field surrounding the particles, it allow us to transform the particle cloud into a polygonal object, unifying the cloud to make it look more like a compact fluid, like a liquid type of fluid.
- Mpolygons: Places polygons facing the outside of the particle cloud.
- Clone object: Places and object from the scene at each particle.

In the next figure, we will present some fluid surface generated by the RealFlow, using the techniques mention earlier in the paper. The fluid is electrically neutral. But using the same technique, only by changing basic fluid equations we are able to simulate the particle movement on electric or magnetic coupled particles (Fig.3.).

Fig. 3. *Fluid simulation using a large amount of particles.*

IV. PROGRAMMING LEVEL

Basically, some of our original applications are developed in *Delphi* (*Pascal for Windows*). Our choose of programming language is basically influenced by the clarity and strong mathematical background with *Pascal*, as well with solid performances and *Windows* environment. Nevertheless, it is not so complicated to move the whole application to C++. Using that basic data structure, all calculations are performed on every quantum separately, using simple numeric form. These procedures are the main part of the engine working in real time. The next procedure calculates movements of the virtual parts of the machines:

```
procedure MOVEt;
var realx,realy,dsr:real;
begin
   for i:=1 to num do
   begin
      realx:=t[i].vx*deltat;
      realy:=t[i].vy*deltat;
      t[i].x:=t[i].x + realx;      {move to x axis}
      t[i].y:=t[i].y + realy;      {move to y axis}
      if (t[i].x<1) then begin t[i].x:=maxx-1; end; {x left limit}
      if (t[i].x>maxx) then begin t[i].x:=1; end; {x right limit}
          if (t[i].y<=1) then begin t[i].vy:=-(t[i].vy/3); t[i].y:=t[i].y
- realy; end; {y control}
          if (t[i].y>=maxy-1) then
                 begin
                   t[i].vy:=-(t[i].vy/3);
                   t[i].y:=t[i].y - realy;
                   dsr:=random(down_speed_reductionn);
                   dsr:=dsr+1;
                   t[i].vx:=t[i].vx/dsr;
                 end;
   end;
end;
```

The wire model (Fig. 5) is based on similar calculations. Based on wire model, we use AutoDesk 3dmax application or some other supporting software to generate high-resolution rendering and animation. We use advanced hardware to meet the extreme requeiments. For instance, simulation and rendering is done on LGA775 Intel Q9550 Core2Quad, 4x2.83Ghz, 1333 FSB, 64 bit Windows XP, 4Gb 1066 GHz memory.

Of course, Tesla fountain is very complex invention (Fig.4), our intention is to model just one of its aspects.

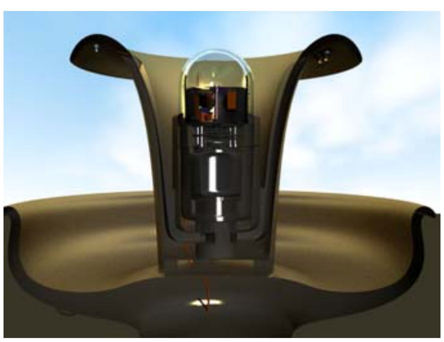

Fig. 4. *Tesla Fountain, the inner structure.*

V. THE PROBLEM OF PARTICLE ROTATION IN 3D VIRTUAL SPACE

One of the very important problem that we solved is how to simulate the three dimensional particle moving in space. The solution must be efficient enough to support real time operations with many thousand particles. This request implies to minimize the using of the trigonometrically (sinusoidal) functions. One of the original solutions that is used in our application is based on different mathematical trigonometrically transformations. The next procedure is very efficient and can be used to generate tree dimensional movements for a large amount of particles in real time. The procedure is written in programming language *Pascal*.

```
procedure rotor(xan,yan,zan,zoom:real);
var   x,y,z,xt,yt,zt:real;              {Temporal coordinates }
sin_xan,cos_xan,sin_yan,cos_yan,sin_zan,cos_zan:real;
    f,sx,sy,zz: integer;
    opti:word;
begin
maximumz:=-30000;

{Trigonometrically transformations of angles in all three
axes.}

sin_xan:=sin(xan);cos_xan:=cos(xan);
sin_yan:=sin(yan);cos_yan:=cos(yan);
sin_zan:=sin(zan);cos_zan:=cos(zan);
for fff:=1 to points_num do
begin
          x:=px^[fff];
          y:=py^[fff];
          z:=pz^[fff];
          Yt:= Y * COS_Xan - Z * SIN_Xan;    {Calculation of
          the position after rotation}
          Zt:= Y * SIN_Xan + Z * COS_Xan;    {around x
          axis}
          Y:= Yt;
          Z:= Zt;
          Xt:= X * COS_Yan - Z * SIN_Yan;  {around y axis}
          Zt:= X * SIN_Yan + Z * COS_Yan;
          X:= Xt;
          Xt:= X * COS_Zan - Y * SIN_Zan;   {around z axis}
          Yt:= X * SIN_Zan + Y * COS_Zan;
px^[fff]:=xt;
py^[fff]:=yt;
pz^[fff]:=zt;
pxx^[fff]:=round(zoom*x+320);    {the zoom factor}
pyy^[fff]:=round(zoom*y+240);
zz:=round(zoom*z);
if zz>maximumz then maximumz:=zz;
pdz^[fff]:=zz;
end
end;
```

This procedure re-calculates all three coordinates from the data arrays, in one pass. The procedure is extremely efficient by using the temporal variables for storage of the trigonometrically function results. The trigonometric calculation is applied on wire models that we generated (Fig.5.).

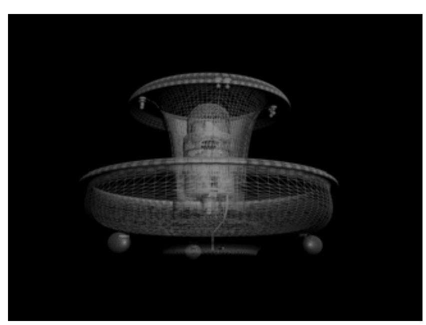

Fig 5. *Wire model of the Tesla's Fountain, applied by the* ***rotor*** *transformation.*

VI. 3D MODELING OF FOUNTAIN

Using original drawings as blueprints during the modeling process the result is a credible three-dimensional object of Tesla's fountain. Three-dimensional objects of all parts of the fountain with the same cross-section from all angles and views are obtained by drawing a half cross-section shape of the object, positioning the axis of rotation and finally applying a *Lathe* modifier on that shape (Fig 6). Lathe modifiers create a 3D model by rotating a shape about an axis of rotation. As almost all parts of the fountain have the same cross-section from all angles and views, the same process of modeling is mainly used.

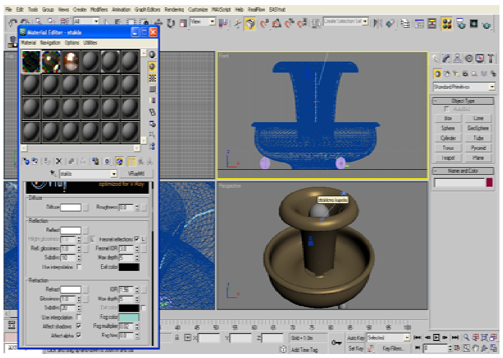

Fig. 6. *3D modeling of the Tesla's Fountain.*

Using our basic model, we are able to applied different materials, to generate for instance a glass model of Tesla's Fountain (Fig. 7.) or internal mechanisms (Fig. 8.).

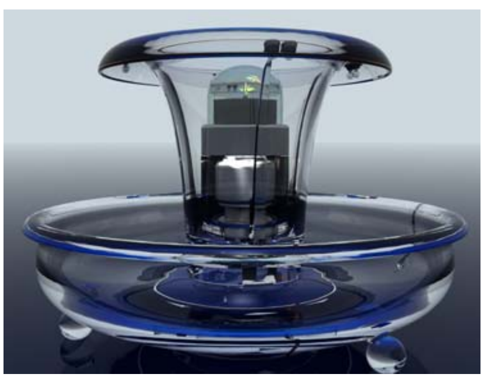

Fig. 7. *Tesla Fountain, model in glass.*

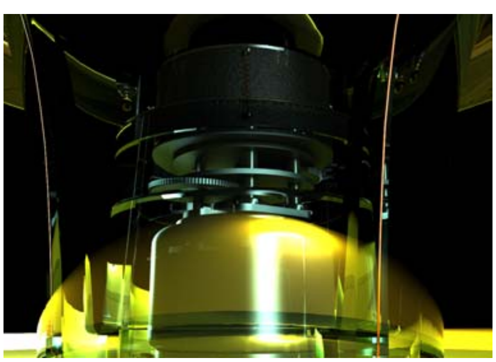

Fig. 8. *3D modeling of the internal mechanism of Tesla's Fountain.*

Using the statically Fountain model, we added the fluids as the particle mash, simulated by the software that we have presented earlier in the paper. By increasing of the number of particles and using the other techniques of representation of the fluid, we generated the next model of fluid-fountain computer model (Fig.9.).

Fig. 9. *Tesla Fountain with fluids.*

Using glass material, we are able to present the internal parts of the fountain, like colour light filters (Fig. 10).

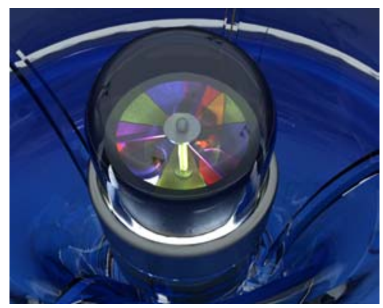

Fig. 10. *Color glass filters.*

VII. CONCLUSION

The basic approach in computer modelling of the complex structures is well known. Some important novelties in this paper are development of the new particle engine by the author that is able to handle a few thousand of particles in real time. Also, some trigonometric algorithms for real time transformation are developed. Some new models of Tesla Fountain are developed and applied many techniques form the particle and fluid modelling to develop complete model of Tesla's Fountain with water as well. The future goal of research that corresponds with project objectives is to develop similar approach to the different branches, for instance for a group of Tesla's energetic packets.

REFERENCES

[1] M. M. Woolfson, *Introduction to Computer Simulation*, OUP Oxford , ISBN-10: 019850425X

[2] Hartmann Alexander *Practical Guide to Computer Simulations,* World Scientific Publishing; Pap/Cdr edition (21 Jun 2009), ISBN-10: 981283415X

[3] M. P. Allen and D. J. Tildesley *Computer Simulation of Liquids,* Clarendon Press; New edition edition (1 Jun 1989), ISBN-10: 0198556454