# Visualization of a Virtual *Caenorhabditis elegans* in WebGL

Andoni Mujika, Gorka Epelde, Alessandro De Mauro and David Oyarzun
*Vicomtech-IK4, Mikeletegi Pasealekua,57, 20009, Donostia-San Sebastian, Spain*

Keywords: WebGL, Visualization, Physically-based Modeling, *Caenorhabditis elegans*.

Abstract: This paper presents the work that has been done in *Si elegans* project in order to visualize the locomotion and the behaviour of a virtual reproduction of the nematode *Caenorhabditis elegans*, one of the most studied animal in neuroscience. The project aims to develop the first hardware-based computing framework that will accurately mimic this worm. It will enable complex and realistic behaviour to emerge through interaction with a rich and dynamic simulation of a natural or laboratory environment. In order to visualize the physical behaviours that emerge from the neuronal system that has been constructed in the project, a web environment has been designed where the user will be able to define an assay and to run it in a WebGL-based 3D virtual arena. For that a relation has been defined from the physics based simulation (run on the server side) and the simplified web rendering of it.

## 1 INTRODUCTION

The *Si elegans* project aims to emulate the neuronal system of a nematode called *Caenorhabditis elegans* (*C. elegans*) (Altun and Hall, 2009). The behaviour that emerges from this emulation will be simulated in a virtual arena, in three dimensions. *C. elegans* is one of the best known organisms in the world and it is widely used, since its genome is completely known and its manipulation for chemical and genetic tests is relatively easy. Moreover, the relative simplicity of its neuronal system (the hermaphrodite *C. elegans* has 302 neurons) and its rich behavioural repertoire (e.g. locomotion, feeding and even certain social behaviours) make this nematode a perfect organism to study and learn from, in the way to understand more complex organisms.

Despite the fact that the connectivity of those 302 neurons is known, the knowledge of the scientific community does not cover every behaviour generated by the neurons and their effects in the worm. That is why *Si elegans* project aims to provide the scientific community with an emulation alternative to the laboratory experiments for their research, accelerating and enhancing the research process.

For that, a web-based three-dimensional virtual arena is being constructed where the user will define the environment (obstacles, air or liquid, etc.), and the behavioural experiments that will be emulated (e.g. touching the worm, placing food or toxic in the plate, applying vibrations to the whole environment). The experiment definition will be transferred to a set of Field Programmable Gate Arrays (FPGAs) where the neuronal emulation will be held. The neuronal network will process and compute the behaviour of the worm and a physics engine will calculate the locomotion of the worm based on it. Finally the physics simulation results will be rendered in the web browser. Additionally, the web-based user interface will display detailed results of the experiment at neuron, muscle and the environment level.

This paper describes the work that has been done in order to connect the web browser and the physics engine so that the user will be able to define the assay easily and will visualize it in a 3D environment. In the following, section 2 reviews the state of art on the *C. elegans* locomotion simulation and its visualization. Section 3 and section 4 describe the physics engine and the web-based interface and finally, section 5 draws conclusions and discusses the future work.

## 2 RELATED WORK

Since early nineties, several works that try to simulate the locomotion of *C. elegans* have been presented. The simulations are becoming more and more realistic, but there is a lot of work to do in order to achieve a perfect reproduction of the worm.

Most of the works that have been done in this

Mujika A., Epelde G., De Mauro A. and Oyarzun D..
Visualization of a Virtual Caenorhabditis elegans in WebGL.
DOI: 10.5220/0005168301640168
In *Proceedings of the 2nd International Congress on Neurotechnology, Electronics and Informatics* (-2014), pages 164-168
ISBN:

field (Suzuki et al., 2005) (Bryden and Cohen, 2008) (Voegtlin, 2011) follow the pioneering work by Niebur and Erdös (Niebur and Erdös, 1991), regarding the shape of the worm and the forces that are applied on it to obtain the locomotion. They construct a 2D reproduction of the nematode with straight segments that represent the muscles. Different forces are applied to those muscles to make the body move in a similar way to the real worm: internal pressure of the worm, elasticity of the cuticle (”skin” of the worm), muscle forces and environmental forces.

Regarding 2D simulation of locomotion, Boyle et al. (Boyle et al., 2012) have been developing one of the most advanced models, where the behaviour of the virtual worm is very similar to the real one in different environments (water, gelatin or agar). Indeed, they showed that the different locomotion of the worm does not emerge from the worm, i.e. its muscles work equally in any environment but the environment makes *C. elegans* move in different ways. They also constructed a neuronal model that makes the worm adapt to the environment with a minimal sensory input.

In recent years, there have been some efforts to upgrade previous 2D works to 3D worlds. Their aim is to offer a perfectly realistic (to go beyond a simple representation in two dimensions) reproduction of *C. elegans* and a easier way to explore its body. The model presented by Mailler et al. (Mailler et al., 2010), using a limited neural model and a worm made by 25 rigid cylindrical sections, obtain a similar movement to those obtained by above described models, but in three dimensions.

The main active project in the field of simulation of *C. elegans* is the OpenWorm Project (OpenWorm, 2014). In this case, the physical model is based on the work of Palyanov et al. (Palyanov et al., 2012), where the virtual worm is composed of point masses and springs to model skin and muscles. The neuronal system of this model uses the 23 neurons that handle forward locomotion (Chalfie et al., 1985). At the moment, in the OpenWorm project they are integrating Predictive-Corrective Incompressible Smoothed-Particle Hydrodynamics (PCI-SPH) (Solenthaler and Pajarola, 2009) to model the behaviour of all the particles that take part in the simulation (including fluids). The realism they want to achieve is very high, but the performance is too slow for the moment.

In regard to visualization of the results obtained by the physical engine described above, the OpenWorm Project has made two different approaches. First, they developed a WebGL-based browser (OpenWormBrowser, 2014), where the user can explore a 3D reproduction of *C. elegans* with all its cells (muscles, neurons and other organs). However, there is no locomotion in this browser. On the other hand, they constructed a web version of their physical engine Geppetto (Geppetto, 2014), where the user can launch some simple particle simulations and watch some neuronal information.

Our work is similar to the OpenWorm project, but there are some differences. Firstly, the neuronal system of OpenWorm is software-based and in *Si elegans*, it is implemented using FPGAs. Regarding physics simulation and its visualization, we aim to obtain a better balance between realism and performance, i.e. we want to obtain a more realistic simulation than other works in the state of the art, without losing usability because of slowness. On the other hand, we can say that we are closer to a complete WebGL-based simulation of the locomotion of *C. elegans* that OpenWorm.

## 3 PHYSICS SIMULATION AND WEBGL IMPLEMENTATION

In this section, we describe the physically-based model that has been defined to make the virtual worm move forward. Besides, we describe the method we follow to take this complex simulation to a web browser.

### 3.1 Physics Engine

Our physically-based model (Mujika et al., 2014) makes a simplification in the muscle structure of *C. elegans*. Usually, the worm has 95 muscles, divided in 4 quadrants of 24 muscles, except for the ventral left one which has 23 muscles. However, in our approach we have considered 96 symmetrically-distributed muscles, as most works described above do.

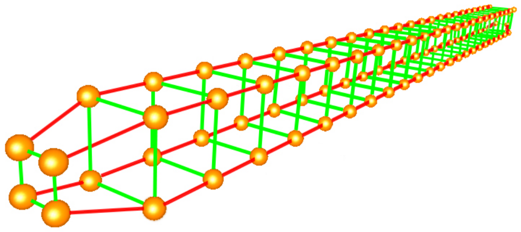


Figure 1: Structure of the virtual *C. elegans* in the physics engine.

In our case, biphasic springs are used to model muscles. A biphasic spring is a regular linear spring that changes its spring constant at a certain length.

This way, changing the stiffness at a certain point we stop the elongation of the spring at certain point. Moreover, in order to generate the locomotion, those springs must be active, i.e. they must contract and relax themselves without any external force. That is why the signal that comes from the neurons (implemented in the FPGAs) is inserted in the equation that rules the behaviour of the springs.

On the other hand, the rings that compound the body of the worm are represented with a square of biphasic springs and are linked to other rings with the muscles. This springs are stiffer and maintain the body shape of the animal. Fig. 1 shows the structure that has been constructed to simulate the worm. In each quadrant, there are 24 red segments that represent the muscles and each ring has 4 green segments. The first and the last ring are smaller in order to obtain a more similar shape to the shape of the worm.

Once the structure of the worm has been defined the forces that will be applied to the masses that are located in the vertices of the structure must be defined.

- As stated before, the elasticity of the cuticle of the worm and the activation of the muscles is obtained by the use of biphasic springs.
- Usual gravity is applied to all the masses.
- As gravity would collapse the whole structure, the internal pressure of the worm must be simulated. For that, a force field that creates a similar effect to a cube made of silicon is defined. The body of the worm is divided into a grid of hexahedra and a Finite Element Method (FEM) is used to simulate the opposition to collapsing tendency of the spring structure.
- Friction is the key that makes the worm go forward and in our model it is based on Signorini's law and Coulomb's friction law. In each time step the colliding points (including those of the worm and those from the floor) are computed. Then, based on the values of forces in the previous time step, an iterative process is done to obtain the actual values of the contact forces.
- Collision with obstacles is detected and the resultant forces are applied to the nematode.

All the physics simulation is run using the Simulation Open Framework Architecture (SOFA), (Allard et al., 2007). It is an Open Source framework primarily targeted at real-time simulation. It offers a scene graph editor for an easy creation of complex environments and it has a wide range of components (solvers, objects, collision detection algorithms, etc.), including the possibility to create new ones.

## 3.2 WebGL Implementation

The simulation of the locomotion of *C. elegans* is carried out in a server using SOFA, but it must be transferred to the web browser. Since most users will be interested in the behaviour of the worm (and not in the muscles themselves), we use a realistic 3D reproduction of the nematode (see Fig. 2) that will be transformed accordingly to the simulation in the physics engine. To render the 3D environment without any plug-in installation, Three.js library (Three.js, 2014), that makes use of WebGL (WebGL, 2014), has been used.

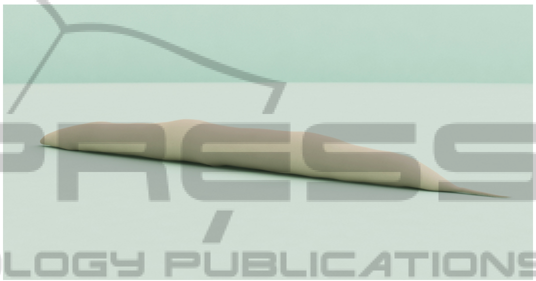


Figure 2: 3D reproduction of *C. elegans*.

Besides rendering the virtual nematode, its movement must be synchronized with the locomotion of the worm in the server. For that, animation bones have been set along the body of the virtual worm. These bones will be transformed accordingly to the mass points of the physics model. The triangle mesh that represents the body is attached to these bones, so the mesh moves accordingly to the transformations of the bones. Fig. 3 shows the positions of the 12 animation bones in respect to the virtual *C. elegans*.

To synchronize the transformations of the 12 animation bones with the muscle movements generated in SOFA, our implementation of SOFA generates a data stream with the positions of the 96 muscles. The WebGL-based 3D environment takes this data and applies it to the animation bones. Specifically, the position of each bone is set to the center of each second ring of the worm in SOFA.

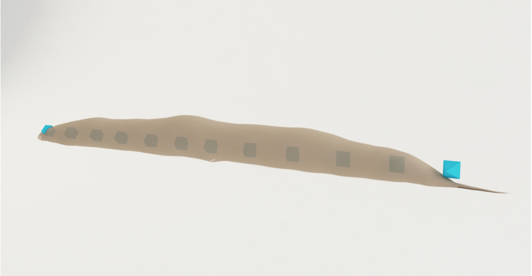
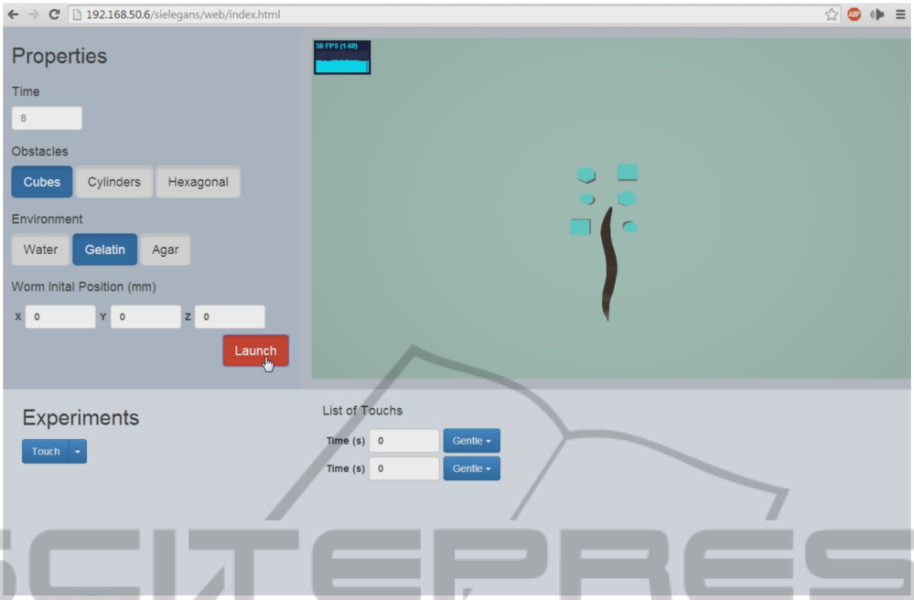


Figure 3: Simulation bones of the worm.



192.168.50.6/sielegans/web/index.html

Properties

Time: 8

Obstacles: Cubes

Environment: Gelatin

Worm Inital Position (mm)

X 0 Y 0 Z 0

Experiments: Touch

List of Touchs

Time (s) 0 Gentle

Time (s) 0 Gentle

Figure 4: 3D simulation of *C. elegans* in web.

# 4 VISUALIZATION

The web browser of *Si elegans* will have a part where the user will define all the aspects related to the experiment he wants to carry out. This part will have three main windows: the environment definition window, the experiment definition window and the 3D window.

In the environment definition window the user will set all the parameters regarding the habitat of the worm. Some parameters will be discrete (defined with a dropdown menu) and other continuous (defined with a numerical value or a slider):

- Shape of the plate (discrete).
- Environmental substance. Discrete (water, air, silicon, etc.) or continuous (density value).
- The initial position of the animal (continuous).
- Friction of the plate. Discrete (different materials) or continuous (friction value).

There may be some aspects that must be defined using both the environment definition window and the 3D window. For example, to insert obstacles in the plate, the user will click in the environment definition window to choose the obstacle type and then click in the 3D window to set the place where the obstacle will be.

On the other hand, the experiment definition window will be used to define what will happen during the simulation. Firstly, the user will have to choose an assay type in a dropdown menu (toxic, feeding, plate tapping, touch) and then the window will show all the options related to this experiment. For example, in a touching experiment, the user will set some touches with their pressure, their timing and the zone where they will be done. As in the environment definition window, some parameters will be set with numerical values (time), with dropdown menus (different pressure levels) and with the aid of the 3D window (zone).

Finally, in the 3D window, the starting point of the simulation (obstacles, plate shape, position of the worm, etc) are rendered using Three.js library. The user will be able to explore the area with the mouse and he will also use this window to define some aspects of the environment or the experiment, as stated before.

Fig. 4 shows the first prototype of the web page, where the user has defined an environment with some obstacles and the simulation is running.

Once the experiment is completely defined, the same 3D window will be used to render the movements of *C. elegans*. As in the experiment definition tool, the user will be able to explore the area with the mouse. At the moment, the prototype only renders the skin of the animal, but in the future, the user will be able to explore all the cells that compound the body.

In the future, the visualization of the behaviour of the animal will be customizable. The user may want to show only some parts of the experiment in the 3D window, watch it in a slower velocity or stop it, etc.

The user will also see the tracing of the neurons that he wanted to be traced. This tracing will have been set in a different user interface. There will be a list of neurons and their corresponding values in a timeline (synchronized with the simulation). The user will be able to remove or add neurons in the list (among those that have been traced) and change the tracing mode (voltage values, only spikes, etc.).

## 5 CONCLUSIONS AND FUTURE WORK

In this paper we have shown our work to simulate the locomotion of the worm *Caenorhabditis elegans* and render it in a web browser without any plug-in installation. On the one hand, we created a physically-based engine that simulates the locomotion of the nematode. The body of the worm is compounded of biphasic springs and different forces are applied to them to obtain the locomotion: muscle forces, elasticity, internal pressure and environmental forces.

On the other hand, we developed a web page where the simulation is shown. The movements of the muscles are transferred to a realistic 3D reproduction of the animal using animation bones. The user can set the parameters of the simulation (environment, position of the worm, obstacles, etc.) and will be able to define the parameters of the experiment (touching moment, toxins, etc.). Once the simulation is defined, the web page will run it and will show the information that the user requires.

Although *Si elegans* project is advancing fast, there is still some work to do. Apart from making the simulation of the locomotion more realistic, regarding visualization, we identify two main lines of work. On the one hand, we want to improve the transference of the information from the physics engine to the web. This way, we will be able to show all the parts of the worm (muscles, neurons, organs) in the 3D environment and offer an innovative way to explore the working of the organism.

On the other hand, we aim to offer neuroscientists a useful tool for the study of *C. elegans* neuronal activity. For that, we have to develop a part in the web page where all the required information will be shown in a friendly and useful way.

## ACKNOWLEDGEMENTS

The *Si elegans* project is funded by the 7th Framework Programme (FP7) of the European Union under FET Proactive, call ICT-2011.9.11: Neuro-Bio-Inspired Systems (NBIS).

## REFERENCES

Allard, J., Cotin, S., Faure, F., Bensoussan, P.-J., Poyer, F., Duriez, C., Delingette, H., and L., G. (2007). Sofa - an open source framework for medical simulation. *Medecine Meets Virtual Reality (MMVR15)*, pages 1318.

Altun, Z. and Hall, D. (2009). Introduction. wormatlas.

Boyle, J. H., Berri, S., and Cohen, N. (2012). Gait modulation in c. elegans: An integrated neuromechanical model. *Frontiers in Computational Neuroscience*, 6(10).

Bryden, A. and Cohen, N. (2008). Neural control of caenorhabditis elegans forward locomotion: the role of sensory feedback. *Biol. Cybern.*, 98:339351.

Chalfie, M., Sulston, J., White, J., Southgate, E., J.N., T., and Brenner, S. (1985). The neural circuit for touch sensitivity in caenorhabditis elegans. *Journal of Neuroscience*, 5:956964.

Geppetto (2014). http://live.geppetto.org.

Mailler, R., Avery, J., Graves, J., and N., W. (2010). A biologically accurate 3d model of the locomotion of caenorhabditis elegans. In *Proceedings of the 2010 International Conference on Biosciences (BIOSCIENCESWORLD10)*, pages 8490.

Mujika, A., de Mauro, A., Robin, G., Epelde, G., and Oyarzun, G. (2014). A physically-based simulation of a caenorhabditis elegans. In *22nd International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, pages 177184.

Niebur, E. and Erdös, P. (1991). Theory of the locomotion of nematodes. *Biophysical Journal*, 60:11321146.

OpenWorm (2014). http://www.openworm.org/.

OpenWormBrowser (2014). http://browser.openworm.org.

Palyanov, A., Khayrulin, S., Larson, S., and Dibert, A. (2012). Towards a virtual c. elegans: A framework for simulation and visualization of the neuromuscular system in a 3d physical environment. *Silico Biology*, 11(3):137147.

Solenthaler, B. and Pajarola, R. (2009). Predictivecorrective incompressible sph. *ACM Trans. Graph.*, 28, 3.

Suzuki, M., Goto, T., Tsuji, T., and Ohtake, H. (2005). A dynamic body model of the nematode c. elegans with neural oscillators. *Journal of Robotics and Mechatronics*, 17(3):318326.

Three.js (2014). http://threejs.org/.

Voegtlin, T. (2011). Clones : a closed-loop simulation framework for body, muscles and neurons. *BMC Neurosci.*, 12.

WebGL (2014). http://www.khronos.org/webgl/.