

Modelling of Flexible Elements

E.V. Afonina¹, M.N. Levaya¹, I.S. Levy¹
cvetelena@rambler.ru|mlevaya@mail.ru|i-levyy@mail.ru
¹Bryansk State Technical University, Bryansk, Russia

Flexible elements are used for modeling of various hawsers, chains, ropes, tail-ropes and other elements in the programmes of simulation. Depending on a current task a set of requirements necessary for a flexible element modeling is formulated. The article presents two methods of rope, wire, cable designing in computer games programming environment Unity3D based on the use of primitive objects. The standard design method grounded in hinge joints employment is considered. The disadvantages of a standard method are noted and two new methods of modeling are offered. The developed methods consider the requirements necessary for the solution of the task at hand, they include the reality of a scene, the interaction with surrounding objects, the possibility to specify the function of a natural sag. Two methods of modeling are compared taking into account the implementation area of programs, the reflection of physical and geometrical properties of modelled entity and the reality of a scene. Both methods meet specified requirements, however it is expedient to use one of the methods for mobile scenes, such as games, exercise machines, simulators of processes, while the other fits better for static programmes.

Keywords: 3D modeling, computer graphics, flexible elements, ropes, cables, wires.

1. Introduction

One of the tasks of rope, chain, hook, knot and other flexible elements modelling is their use in the development of 3D-games, trainer simulators and simulators, testing and training programmes, commercials and other animation projects.

Flexible elements such as ropes and tail-ropes give interesting and fun dynamics to game scenes. They can be used for moving through levels, battles taking place in arenas and also for retrieving items. In various specified 3D simulators flexible elements can be employed to simulate wires and cables of overhead power lines or various devices for both carrying out rescue operations and, for example, lifting cargos. In one form or another, flexible elements are used practically in all 3D scenes.

In the study of the mechanical properties of wires, ropes, cables, geometric models do not raise a demand of reality. Depending on the scope of application and study of the model, modeling methods can vary significantly [1, 3-4].

2. Existing methods of modeling flexible elements

By "flexible element" we shall basically mean any segment, the main characteristics of which are length, strength and flexibility. These include hawsers, ropes, chains, wires, cables and other elements, which can connect two or more objects. To give realism to the scenes flexible elements should reflect physical properties of real objects, for example, oscillation under impulse load, natural sagging at the horizontal orientation of a segment, etc.

So far a certain method of creating ropes and tail-ropes in three-dimensional modeling programmes has been developed. Modeling of games, feature and instructional films, 3D characters of advertising are performed in Autodesk 3ds Max, Cinema 4D, Sculptris, Icon, Sketch Up, Sweet Home 3D, Visicon, Paint 3D, Blender 3D and others [2, 5, 6]. But the functionality of each of these programs is well designed to solve certain types of tasks, and is not suitable for modeling of specialized scenes due to its inability of either reflecting physical properties of such flexible objects like wires and chains or interacting with other elements of the scene.

When creating production processes of training simulators and simulators, building of ropes, tail-ropes, wires, models should meet the following requirements:

1. reflection of physical properties (elasticity, flexibility, oscillation in motion, etc.);
2. realistic appearance of modeling object;

3. a small number of polygons, which is good for high performance of 3D scene modeling.

For clarity of the flexible element movement simulation the cross-platform programming environment of computer games (the so-called game engine): Unity3D was chosen. The main advantages of Unity3D include the presence of a visual development environment, cross-platform support and modular system components. For nontrivial tasks, native components written in the C# or JavaScript programming language are used. The difficulties appearing when working with multicomponent schemes and problems in connecting external libraries, both leading to a sharp drop in performance, can be considered as disadvantages.

To solve this problem Unity 3D has a standard mechanism based on the use of the component HingeJoint.

HingeJoint is a component that connects two objects tipping the scale and allows them to rotate around a specific point or axis. When creating a flexible element, primitive objects like spheres and capsules were used. Each sphere connects two capsules using the HingeJoint component. At the two ends of the flexible element there are posts with the specified IsKinematic property. It is assumed that they will not be affected by the force of the flexible element. These posts are also connected to the extreme capsules by a hinge joint.

As a result of modeling, we obtain a flexible element presented in Fig. 1.

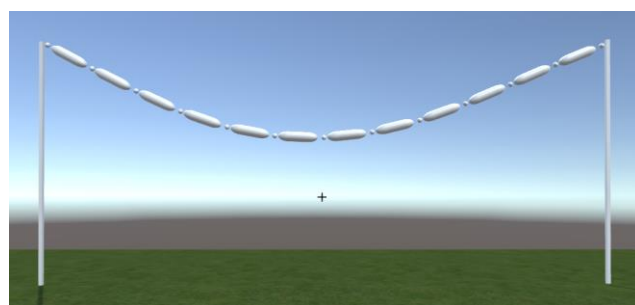


Fig.1. Simulation of a rope (tail-rope) using the HingeJoint component

This approach has its drawbacks:

1. The employment of primitives (spheres and capsules), that makes creating of real objects difficult.
2. The uprising of gaps between primitives appearing at the maximum amplitude of oscillation.

3. The inability of element staying. Less sagging of the element requires the change of the primitives' number, and the display quality of the flexible element is lost.

In the development of a new method of flexible elements modeling, the set task is to avoid these disadvantages.

3. Methods of modeling flexible elements taking into account specified requirements

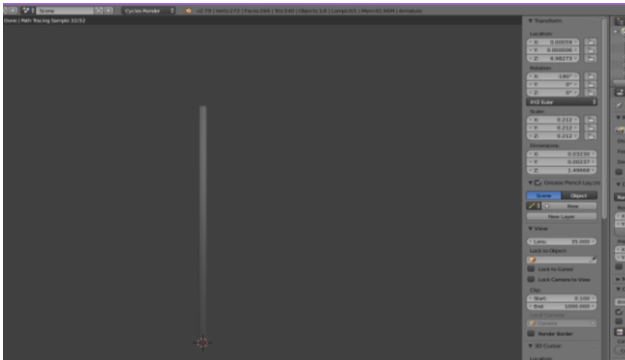
The first of the possible methods of designing lies in the fact that flexible element should be a single mesh. Mesh or baseline grid is a term used to denote a total of top elements, edges, and polygons that make up a single 3D object. A rope object is created in any 3D editor using primitives, for example, cylinders. Then tools of object deformation by means of armature (skeleton) are employed. The more "bones" the skeleton has, the more the simulated object will resemble the movement of a flexible element. By using this method it is possible to get rid of the first two disadvantages described above, but at the same time it requires bringing in additional programmes.

The second method is connected with using the standard LineRenderer component of the Unity3d programme. This component consists of an array of two or more points of three-dimensional space and successively draws a straight line between each pair of points (that is, at three or more points a polyline is obtained), therefore the created object looks seamlessly (integrally). The use of this method makes the simulation of the physical properties of the object possible.

To implement the first method we take advantage of 3D editor of the Blender graphics package. Blender 3D editor has enough feature set to create 3D models, videos with complex scenes, sculpturing and creating realistic object visualizations.

To model a rope (tail-rope) a standard cylinder object should be designed, which is then split into parts in edit mode (Fig. 2).

Fig.2. Rendering of flexible element.



The next step is to add a standard object "armature" (skeleton) to the project and create a child object of CAD fittings ("bone") for each part of the cylinder (Fig. 3).

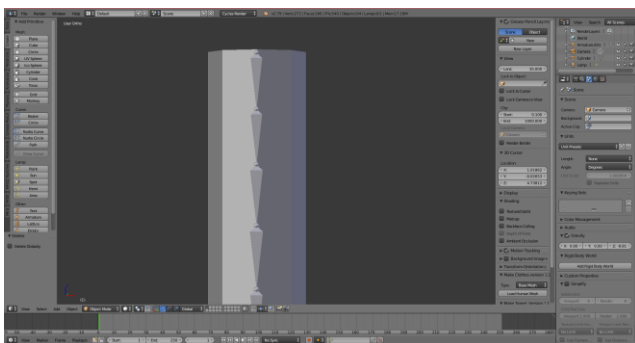


Fig.3. CAD fittings (armature) of flexible element.

Going forward it is imperative to bind the "bones" of the armature to the designed object. The resulting object can freely rotate its parts around three axes. For clarity it is necessary to

enter the posing mode, and set the "skeleton of bones" rotation about different axes (Fig. 4).

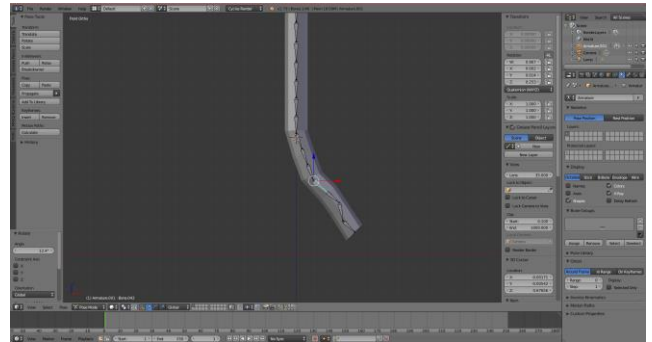


Fig.4. Mode of the flexible element posing.

To simulate the motion of a flexible element the C#script component is added to the created object. The model is then imported into Unity3D.

The following components must be specified for each part of the flexible member:

1. Rigidbody (a solid body).

With Rigidbody, objects can interact with each other taking into account physical properties by means of the programme describing physical state, titled NVIDIA PhysX. For the body, you can set its mass, the value of gravity, the forces acting on it, the torque.

2. SphereCollider (collider). The SphereCollider component determines the shape of an object (in this case it is the shape of a sphere), which is necessary for calculating mechanical collisions with other objects. This component is transparent, and the geometric shape may not match the exact shape of the object.

3. DistanceJoint3D. This designed component allows calculating physical properties of the interaction between objects, such as speed (velocity), distance (distance), the force that must be given to the object to change the position (spring), the attenuation of the object during its perturbation (damper), etc.

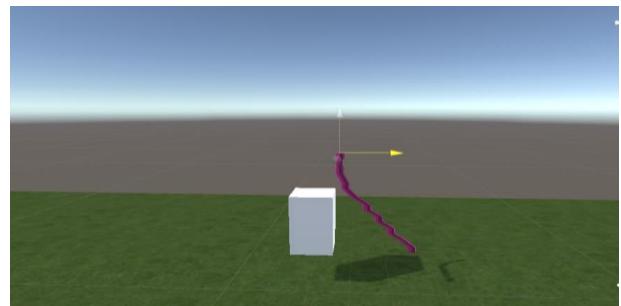
4. The "private void AddChildren (Transform parent)" function adds created components for each part of the flex element. The calculation of physical properties of the relationship between the parts of the rope is made separately in the "void FixedUpdate" function of the "DistanceJoint3D" component, which is called each time when the simulation scene frame is updated.

As a result, we obtain a simulation of a flexible body on the scene shown in Fig. 5.

Fig.5. Modeling of the physical properties of a flexible element.

Modeling by using the second method is aided by standard component LineRenderer (render lines) programmes Unity3d.

To do this, an empty object is added to the scene and the



LineRenderer component is "attached" to it.

To simulate the motion of a flexible element, a C#script is added to the object. In the developed script, you can change the following properties:

1. Pointsinlinerenderer: the number of points in the LineRenderer component defines the rope (the smaller the number of points, the greater the performance of frame-by-frame rendering of the rope physics).

2. Sag Amplitude is a property that determines how much the rope will sag in the middle.

3. Sway Multiplier is a property that estimates how much the rope will deviate in different coordinates.

4. Sway X Multiplier is a deviation of the rope in the local coordinate system by the x coordinate.

5. Sway Y Multiplier is a deviation of the rope in the local coordinate system by the Y coordinate.

6. Sway Frequency is a property that determines the frequency of the rope vibrations.

7. End Point Transform is a property that determines the coordinates of the object to which the other end of the rope will be attached (it is desirable for the object being empty).

The coordinates of the points of the LineRenderer component are calculated separately in the "void Animate" function, which is called each time when the simulation scene frame is updated. The calculation is made taking into account all the above-mentioned properties.

The result is a simulation of the physical properties of the flexible element in the scene shown in Fig. 6 with the amount of sagging of 5- Unity meters and in Fig. 7: 1Unity- meter.

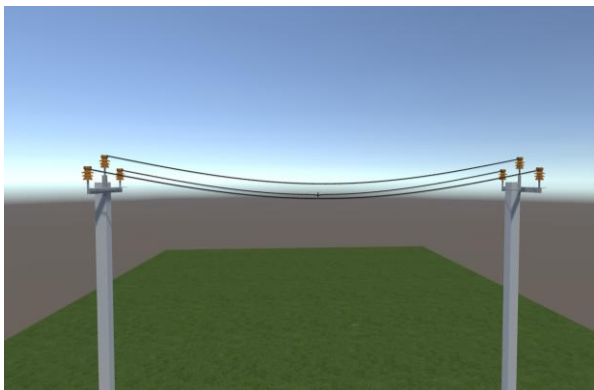
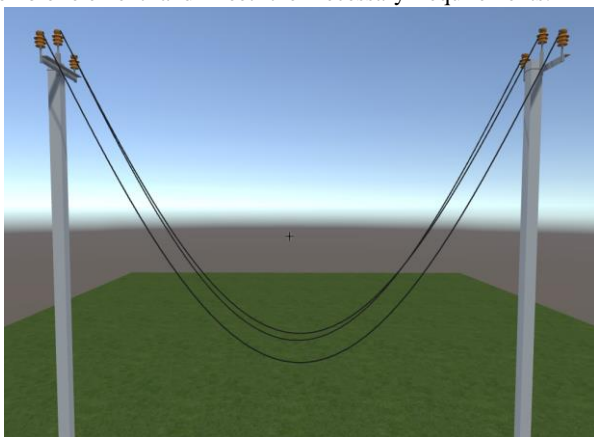


Fig.6. Simulation of the physical properties of a flexible element with a sagging of 5 Unity-meters

Fig.7. Simulation of the physical properties of a flexible element with sagging at 1 Unity-meter.

As a result, both developed methods eliminate the disadvantages of the standard modeling mechanism of the flexible element and meet the necessary requirements. The



designed objects appear as a whole. There is also a range of settings for physical properties adjusting. When specifying a particular type of flexible element such as wire, rope, tail-rope

and other it is required a certain texturing to the model [7, 8] to create the vividness of the scene.

4. Conclusion

The developed methods meet the specified requirements, have positive features and preferential scope of use [1, 9, 10].

A significant advantage of the first method is the presence in all parts of the object of the rope collider component, which coincides with the shape of the rope. This makes the interaction with other objects in the scene easier. The interaction is carried out by means of add-in physical properties determinant of the Unity 3D programme with a number of settings.

This method is useful for ropes (or other flexible elements) that often interact with other objects in the scene, such as catapult tension harnesses (in games, simulators, process simulators, etc.).

The advantage of the second method is a simpler calculation of the rope movement, since a single object is used to form a flexible element, which contributes to the programme efficiency. Besides it there is no need to use additional programmes to build the rope. This method is good in static scenes, as it is shown in the example of high-voltage power lines.

5. References:

- [1] Voloboy A., Galaktionov V. Visualisation technologies in the problems of modern industry//Scientific visualization.- 2013.- Vol.5. № 4.-pp. 31-46.
- [2] Ermakov E., Mitin S., Solving kinematic problems of models and units of mechanisms under geometric constraints [Article] // GraphiCon.- 2016 [Proc. of the international scientific conf.]; Russia, Nizhny Novgorod, Sept.,19–23,2016- pp. 77-79.
- [3] Zakharova A.A., Vekhter E.V., Shklyar A.V. Methods for solving data mining tasks when using analytical visual models // Scientific visualization. – 2017. – Vol. 9. – № 4. – pp. 78-88.
- [4] Bednarczyk B.A., Stier B., Simon J.-W., Reese S., Pineda E.J. Meso- and micro-scale modeling of damage in plain weave composites. Composite Structures. 2015. Vol. 121. pp. 258 – 270.
- [5] Behera B.K., Dash B.P. Mechanical behavior of 3D woven composites. Materials & Design. 2015. Vol. 67. pp. 261 – 271.
- [6] Brodlie, K.; Osorio, R.A.; Lopes, A. A review of uncertainty in data visualization. In Expanding the Frontiers of Visual Analytics and Visualization; Springer: Berlin / Heidelberg, Germany, 2012; pp. 81–109.
- [7] Geuzaine C., Remacle J.-F. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and postprocessing facilities // International Journal for Numerical Methods in Engineering, Vol. 79, i. 11, pp 1309-1331, 2009.
- [8] Kirby R., Silva C. The need for verifiable visualization // IEEE Computer Graphics and Applications, 28(5) Sept. 2008. pp. 78–83.
- [9] Parsons E.M., King M.J., Socrate S. Modeling yarn slip in woven fabric at the continuum level: Simulations of ballistic impact. Journal of the Mechanics and Physics of Solid.- 2013-. Vol. 61.pp. 265 – 292.
- [10] Zakharova, A.A., Shklyar, A.V. Visual presentation of different types of data by dynamic sign structures. Scientific visualization Vol.8., № 4 ,pp.28-37 (2016).