International Journal of u- and e- Service, Science and Technology
Vol.9, No. 3 (2016), pp.179-188
http://dx.doi.org/10.14257/ijunesst.2016.9.3.18

A Design and Simulation of 3D Fishing-Net Considering Tidal-Current and Buoyancy

Joseph Yoon[1], Oh-Seok Kwon[2], Ga-rim Yoon[3] and Young-Bong Kim[4,*]

[1,2,3,4]*Pukyong National University*
[1]*ricerich@gmail.com*, [2]*inartistical@gmail.com*,
[3]*garimyoon@hotmail.com*, [4]*ybkim@pknu.ac.kr*

Abstract

A physically based simulator for a cloth or patch has often utilized a mass-spring model. This mass-spring model generally employs the gravity force. To make a simulation model of fishing-net, we have to consider the tidal-current and buoyancy in underwater environments. This additional factors lead to get a lower calculation speed for a real-time simulation. In this paper, we will propose a new mass-spring model for a fishing-net and also a method to simplify the equations for the real-time simulation of a fishing-net model. Our 3D mass-spring model will show the mesh-structure similar with the other mass-spring model except that each intersection point can have independently a different mass. The motion of each mass will be calculated with adding dynamics. To reduce the calculation time, we will try to simplify the mathematical equations which consider the tidal-current and buoyancy. Through this research, we can get the real-time and realistic simulation for the fishing-net.

Keywords: *simulation, fishing-net, tidal-current, buoyancy*

1. Introduction

A simulation of 3D fishing-net has been used to check the efficient spread of a net and also exercise the fisherman. These simulation systems have often employed the mass-spring model to simulate the cloth or patch because of the shape of fishing net.

In such simulation systems, the calculation speed is one of the critical issues to get real-time simulation [1-2]. There have been developed many researches to increase the calculation speed. Ko[2] proposed the method to reduce the integration time. Lee[3] simplified the equations to solve the motion of fishing-net using constant term. These previous researches have basically considered the only gravity factor to simulate the motion of mass-spring model. However, the fishing-net has to apply the factors such as tidal-current and buoyancy to determine its motion. And also it needs to consider the non-uniform mass distribution of a fishing-net because a fishing-net can be formed with several netting twines with different weights. These additional factors make the simulation system difficult to get the real-time simulation because we have to create a model to include the factors like the density of fluid, the density of object, the volume of object, and the vector of tidal current.

Therefore, we propose a practice system for the 3D fishing-net which is deduced from general mass-spring systems. We obtain an accurate motion of fishing-net using a two-step approach: *Input Data Control* that get actual fishing-net data with different masses. *Physics Processor* that calculates the motions from tidal-current, buoyancy, and the gravity.

In our system, we will generate a reality simulation in specific undersea environments. We first make the *Input Data Control* panel that gives more actual data input and easier-to-use UI. Our *Physics Process* will generate the better real motion than the motion with only gravity force. These additional forces increase the calculation speed because the

ISSN: 2005-4246 IJUNESST

mathematical equations require the more complicated computer calculations. In order to reduce the calculation time, we will simplify the mathematical equations considering the tidal-current and buoyancy. This calculation will provide the similar motion with a real fishing-net model.

In Section 2, we will provide the system overview to make the real fishing-net simulator which expresses the various forces such as tidal-current, buoyancy, and the gravity. Section 3 shows the data input panel to represent the structure of the fishing net. The motion of the fishing-net will be calculated with a physics processor in Section 4. Conclusion remarks will be given in Section 5.

2. System Overview

Our system has three steps as shown in Figure 1. In first step, *Input Data Control* inputs the different mass as well as several properties into the structured fishing-net data. *Interactive User Interface* determines the external forces such as the lift force of tidal-current, the density of object, the volume of object from keyboard. It also initializes a position vector of fishing-net by user's mouse control. Second step is the *Physics Processor* which calculates a fishing-net motion by the characteristics of fishing-net structure such as tidal-current, and buoyancy.

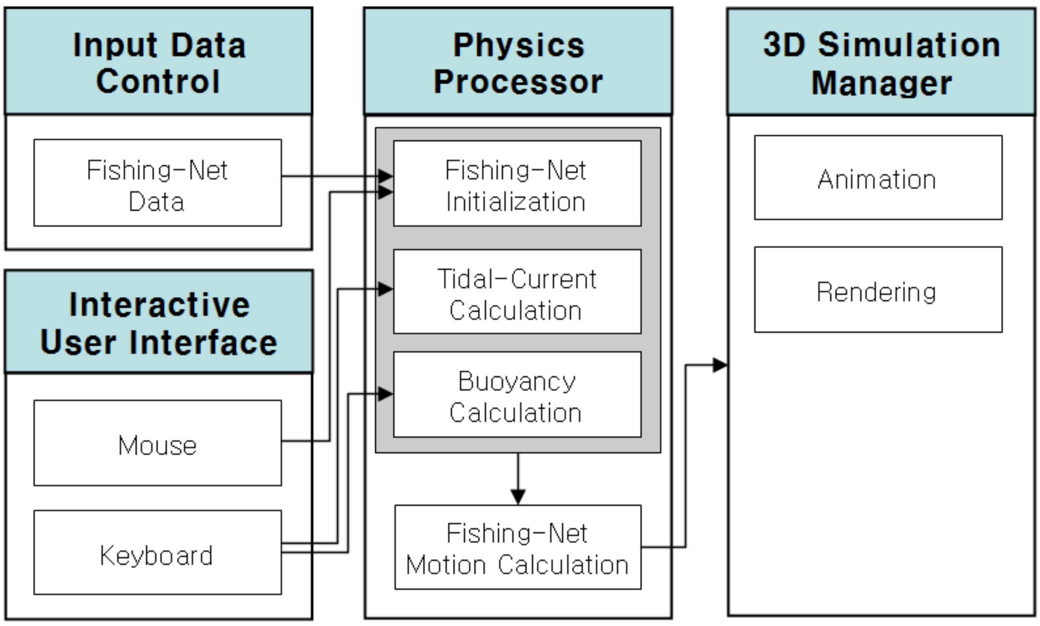

Figure 1. System Overview

Last step, *3D Simulation Manager*, shows the natural motion on the display monitor using computed values.

3. Input Data Control

3.1. Fishing-Net Data

Fishing-net data affect the reality of the fishing net that we try to express in our fishing-net simulator. So we have to use actual data to implement our fishing-net model.

As shown in Table 1, each point should take various factors that well represent the actual fishing-net properties.

Table 1. Input Data of Fishing-Net

No	Symbol	Definition	No	Symbol	Definition
1	id	ID of point(=object)	8	surface	Projected area
2	x	Position of x	9	part_type	Type of point(=object)
3	y	Position of y	10	link_count	Count of link
4	z	Position of z	11	link_id [ ]	ID list of links
5	volume	Volume of point(=object)	12	l	Length
6	density	Density of point(=object)	13	k	Tension
7	mass	Mass			

General mass spring model often use the rectangle structure that connects the neighboring points in a rectangular shape.

However, the real fishing-net shows many diamond shapes and vertical lines. We also construct the mass-spring model of the fishing-net such as following Figure 2. Our fishing-net model uses the mixture of diamond structures and triangle structures.

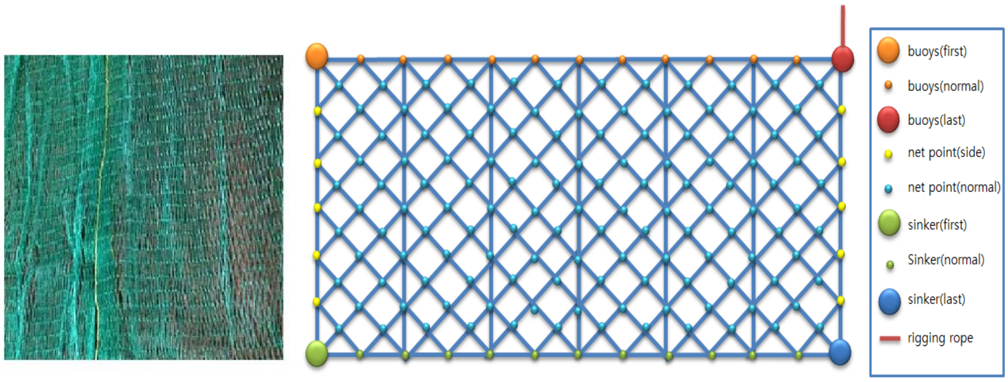

Figure 2. Mass-Spring Model of Fishing-Net

The fishing-net model consists of finite number of mass points and each intersection point is connected with its neighboring points through the spring. As shown in Figure 2, each point can have the different density, and it is classified into three types. First type is the buoys located at top-line and make the fishing-net lift. Second type is the sinker located at bottom-line that fall down the fishing-net. Normal points located at middle region are the third type. Each point has different density values depending on its position in a fishing-net. Each spring can own different spring coefficient because it plays a different role in the fishing-net.

4. Physics Processor

4.1. Fishing-Net Initialization

In a *Physics Processor*, the initial fishing-net model can be formed using the actual net information from *Input Data Control*, and a position of fishing-net through *Interactive User Interface*. First, a structured data for mass-spring model consists of *NetData*, *NeighoborData*, and *SpringData* as shown in Figure 3. In addition, *RenderData* uses finally to display fishing-net.

```
struct  NetData {
struct  NetVertex{
int       id;
float     posX, posY, posZ;
float     volume;
float     density;
float     mass;
float     area;
float     surface;
};
int             vertexID;
NetVertex  *netVertex;
};
```

```
struct   SpringData {
struct   SpringVertex {
         int      selfIndex;
         int      linkCount;
         int      *linkingNums;
         float    *lengths;
         float    *tensions;
};
int    srpingNum;
SpringVertex  *springVertex;
};
```

```
struct NeighborData {
struct  NeighborVertex{
        int    id;
        int    counts;
        int    *polygonIndex;
};
Int     vertexID;
Int     accumulateID;
NeighborVertex  *neighborVertex;
};
```

```
struct   RenderData {
struct   RenderVertex{
         float    posX, posY, posZ;
         float    normalX, normalY, normalZ;
         float    velocityX, velocityY, velocityZ;
         float    colorX, colorY, colorZ;
};
RenderVertex    *renderVertex;
};
```

Figure 3. Structured Data for Mass-Spring

NetData includes net's properties such as id (=vertex id), position, and mass at each points, and *NeighborData* has related information on connecting points such as id(=vertex id), counts(=total count of link), and accumulate id. Then, *SpringData* generates mass-spring model connecting two adjacent net-vertices, and it includes link count (=total count of linking spring), length (=length of spring), and tension (=tension of spring). The characteristics of the fishing-net motion depend greatly on the position vector of points. Therefore, in *Interactive User Interface*, we can handle to move each point after picking a vertex. This changing value can be the other input value into *Fishing-Net Initialization*.

4.2. Motion Equation in a Fishing-Net

The mesh in an actual fishing-net should be approximated with a small number of mass points to get a rapid calculation. So, we first have to reduce the mass points of a fishing-net model in allowable range. At each mass point on the fishing-net, the motion equation is:

$$F_{(t)} = F_g + (F_c + F_{cd} + F_{cl}) + (F_b + F_{bd}) \tag{1}$$

Where F_g is a gravity force, $(F_c+F_{cd}+F_{cl})$ is a tidal-current force, and (F_b+F_{bd}) is a buoyancy force.

4.3. Adopting Tidal-Current Force

Tidal-current force is defined as F_c, F_{cd}, F_{cl} in above motion equation. F_c is the vector of the tidal-current, and F_{cd} is the drag force that the tidal-current affects on the object. F_{cd} is calculated as follows:

$$F_{cd} = S * \hat{C} * (1 - (N \bullet \hat{C})^2) \tag{2}$$

Where S is the projected area of the object, $\hat{C}$ is normalized result of F_c minus the velocity of the object, and N is the normal vector of the object.

$$\hat{C} = \frac{F_c - v}{\left\| F_c - v \right\|} \tag{3}$$

F_{cl} is the lift force that the tidal-current makes the object afloat. F_{cl} can be defined as follows as:

$$F_{cl} = S * V_L * K_L \tag{4}$$

Where S is the projected area of the object, V_L is the vector of the lift force, and K_L is the coefficient of the lift force. V_L and K_L are given as follows:

$$V_L = N - \hat{C} * (N \bullet \hat{C}) \tag{5}$$

Where N is the normal vector of the object, v is the velocity of the object, and F_c is the vector of the tidal-current. From above equation, we obtain the vector of the lift force as V_L.

$$K_L = L_{func}(1 - 2(N \bullet \hat{C})^2) \tag{6}$$

Where K_L is a very important factor because it determines scale of lift force. In order to generate K_L, we should define a lift function $L_{func}(\ )$. The function needs the parameters such as N, F_c, and v . The N is the normal vector of the object, F_c is the vector of the tidal-current, and v is the velocity of the object. In order to get real motion of fishing net model, it is necessary to get good function $L_{func}(\ )$. We will simplify this lift function through the adequate mathematical formulation to approximate discrete experiment results given in Table 2.

Table 1. Coefficients of Lift Force Depending on Angle

No	Angle	Coefficient of Lift force	No	Angle	Coefficient of Lift force
1	0.00	0.020675	13	51.00	0.504986
2	5.00	0.029830	14	55.00	0.517126
3	6.00	0.028353	15	56.00	0.521239
4	10.00	0.052438	16	60.00	0.496089
5	11.00	0.049820	17	61.00	0.500870
6	20.00	0.138550	18	65.00	0.427470
7	21.00	0.136940	19	66.00	0.435763
8	30.00	0.265190	20	72.00	0.238003
9	31.00	0.265750	21	73.00	0.235445
10	43.00	0.438975	22	85.00	0.088935
11	44.00	0.441512	23	86.00	0.092204
12	50.00	0.501985	24	90.00	0.057054

Using above data, lift function $L_{func}(\ )$ can show the curvature graph as shown in Figure 4. The lift function has the peak value around 55.00 degree and also the lower value at 0.0 degree and 90.0 degree. Through this graphical property, we can know that the shape of lift function resembles the typical cosine function.

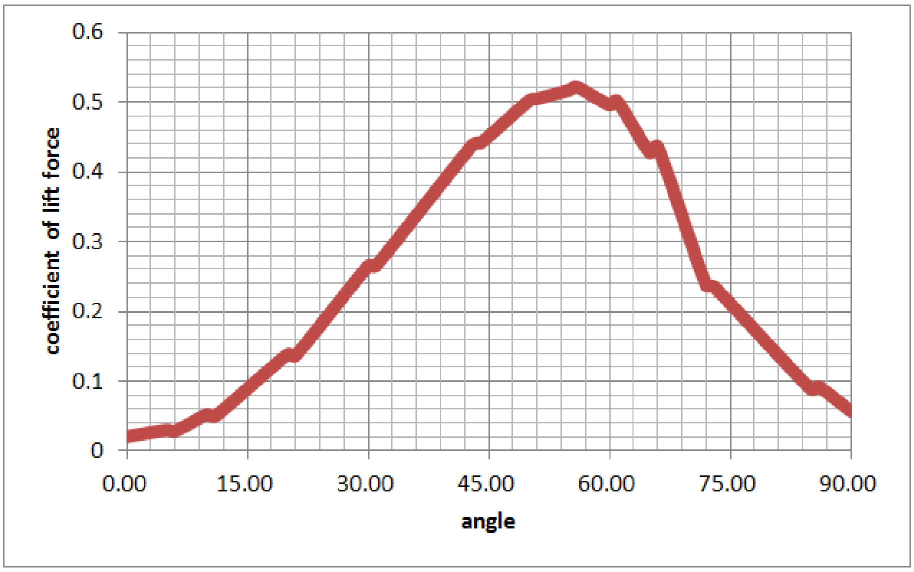

Figure 4. Graph of Lift Function, $L_{func}(\ )$

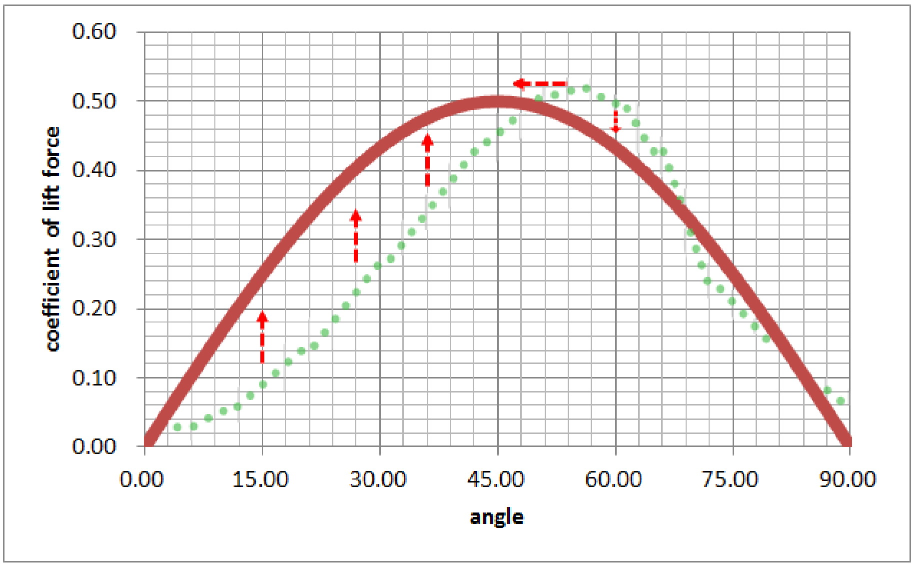

Figure 5. Comparison of $L_{func}(\ )$ and *Cos ()*

Figure 5 shows the difference of the lift function and the typical cosine function. The difference represents a maximum value around 30.0 degree. The value is about 70% greater than an experimental data. From these characteristics, we assume that the lift function can be replaced with the cosine function. $L_{func}(\ ) \cong cos(\ )$ can be approximated as follows:

$$K_L = cos(1 - 2(N \bullet \hat{C})^2) \tag{7}$$

The typical cosine function is a very time consuming job. So, we propose the work to determine the cosine result of a given value using Taylor series.

Thus, the coefficient of lift force, K_L, can be compensated using Taylor approximations as follows:

$$K_L = Taylor(1 - 2(N \bullet \hat{C})^2) \tag{8}$$

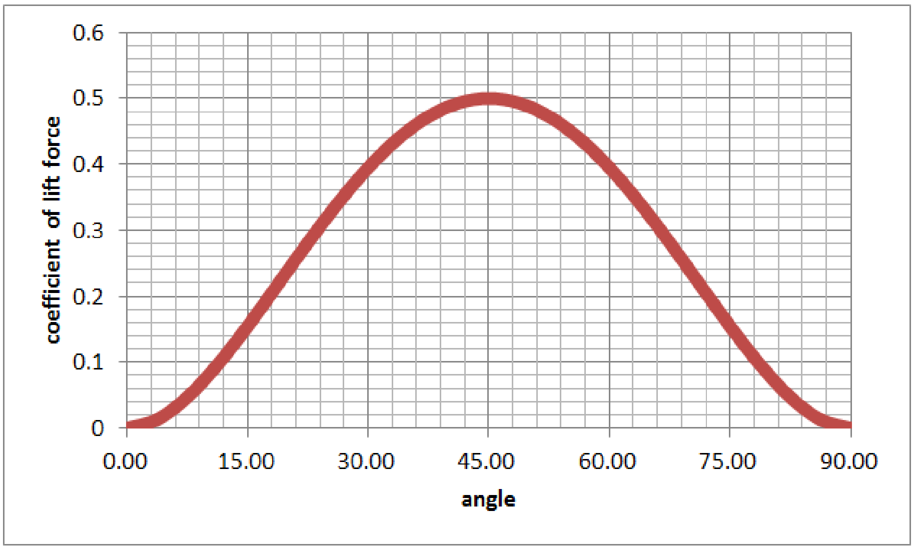

Figure 6. Compensation using Taylor Approximations

In our system, as shown in Figure 6, we obtain the value K_L using the second order Taylor series for the calculation speed of fishing net model.

4.4. Adopting Buoyancy Force

Buoyancy force is expressed as F_b, F_{bd} in the motion equation. If F_b is the buoyancy force, D_o is the density of the object, D_f is the density of the fluid, V is the volume of the object, and W is the weight of the object,

$$F_b = (D_o - D_f) * V * W \tag{9}$$

The density D_f affects the great effects on the motion of fishing-net. In specific undersea environments, D_f has different value according to depth of water as shown in Figure 7.

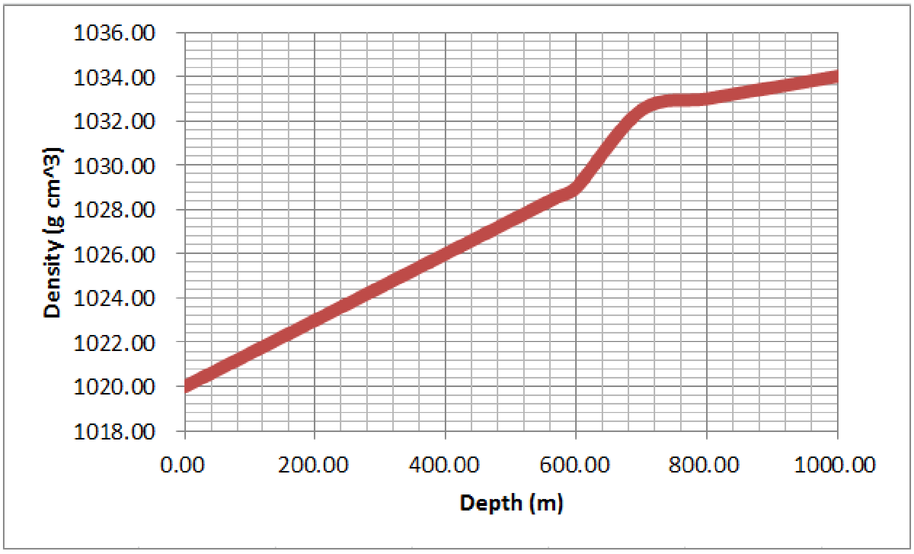

Figure 7. Density of Fluid in Undersea

However, above depth density makes great expense in computing because it is a nonlinear. So, we simplify the depth density through linear approximation as shown in Figure 8:

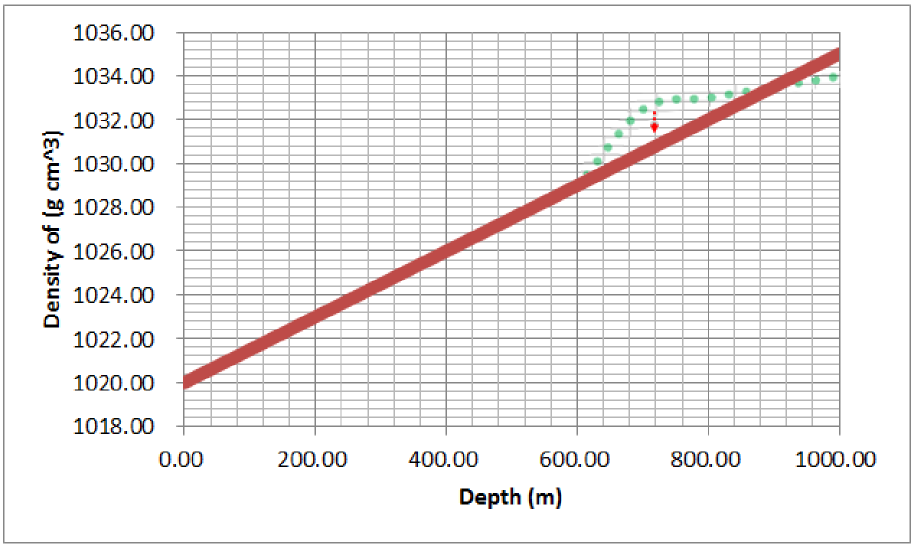

Figure 8. Simplification of Linear Approximations

Therefore, the D_f is determined through above linear simplification.
In our system, the drag force of buoyancy is defined as follows:

$$F_{bd} = D_f * S * K_b \quad (10)$$

Where D_f is the density of the fluid, S is the projected area of the object, and K_b is coefficient of the drag force.

5. Experimental Results

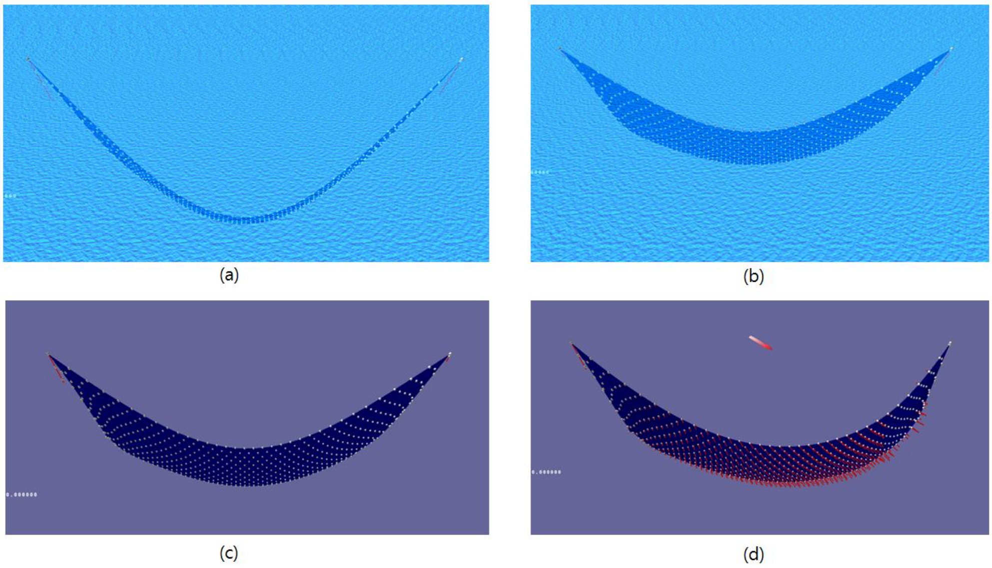

Figure 9. Result of Simulation

We implemented the 3D fishing-net simulator using described techniques. Figure 9(a) is a result that did not consider the tidal-current and buoyancy force. The middle points are drooped because the gravity is adopted on the mass points. Figure 9(b) shows the effects of the buoyancy. The buoys at top lines make the fishing-net lift and the sinks at the opposite side falls down the fishing-net. Figure 9(d) is the result image that the tidal-current force is added to the Figure 9(c). In Figure 9(d), we can observe that fishing-nets are moved to the right-bottom direction. The amount of shift depends on the value of tidal-current force.

6. Conclusion

In this paper, we have developed a 3D fishing-net simulation system considering tidal-current and buoyancy. The physics process is implemented to integrate the physics model as well as to calculate math model. The mathematical equations are simplified to get a real-time simulation. The experimental results show that our simplification of mathematical equation to compute the motion of fishing-net is very efficient and the UI panel to input the characteristic data of fishing-net is convenient and simple. Through this research, we can make a base technique to get the real-time and realistic simulation for the 3D fishing-net.

Acknowledgments

This work was supported by a Research Grant of Pukyong National University(2015 year).

References

[1] D. Baraff and W. Andrew, "Large steps in cloth simulation", Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques(SIGGRAPH), Orlando, USA, July **(1998)**, pp. 19-24.

[2] K. Choi and H. Ko, "Stable but responsive cloth", The 32nd International Conference on Computer Graphics and Interactive Techniques(SIGGRAPH2005), Los Angeles, USA, **(2005)**, July 31-August 4.

[3] G. Lee, "Development of fishing simulation model and implementation of fishing simulator", Pukyong National University (Thesis of Doctor), Busan, **(2009)**.

[4] J. Lee, L. Karlsen and C. Lee, "A method for improving the dynamic simulation efficiency of underwater flexible structures by implementing non-active points in modelling", ICES Journal of Marine Science, vol. 65, no. 9, **(2008)**, pp. 1552-1558.

[5] R. Bridson, R. Fedkiw and J. Anderson, "Robust treatment of collisions, contact and friction for cloth animation", ACM Transactions on Graphics (ToG), vol. 21, no. 3, **(2002)**, pp.594-603.

[6] S. Olga, C. Daniel, L. Yaron, A. Marc, R. Christian, and S. Hans-Peter, "Laplacian surface editing", Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing, Nice, France, **(2004)** July 8-10.

[7] H. Y. Kim, C. W. Lee, J. K. Shin, H. S. Kim and B. J. Cha, "Verification of mathematical model on purse seine gear through sea trials and dynamic simulation", Proceedings of the 7th international workshop, Busan, Korea, **(2005)** November 23-26.

[8] K. Hwang, J. Lee and S. Jang, "Buoyancy-driven heat transfer of water-based Al_2 O3 nanofluids in a rectangular cavity", International Journal of Heat and Mass Transfer, vol. 50, no. 19, **(2007)**, pp. 4003-4010.

[9] J. S. Turner, "Buoyancy effects in fluids", Cambridge University Press, **(1979)**, London.

Authors

Joseph Yoon, he received the B.S., M.S. and completed Doctor Course in Computer Science from Pukyong National University, Korea in 2003, 2005, and 2007 respectively. Since then, he had worked as developer with Sensingmap Co., Ltd. at Pusan National University in Korea until 2011. Since 2012, he has been studying with Department of IT Convergence and Application Engineering at

Pukyong National University in Korea. His main research interests include Computer Graphics and Human-Computer Interaction.

Oh-Seok Kwon, he received the B.S., M.S. in Computer Science from Pukyong National University, Korea in 2012 and 2014 respectively. Since 2015, he has been Doctor Course Student with Department of IT Convergence and Application Engineering at Pukyong National University in Korea. His main research interests include Computer Graphics, GPGPU and Biomedical Image Visualization.

Ga-Rim Yoon, she received the B.S., M.S. and completed Doctor Course in Computer Science from Pukyong National University, Korea in 2003, 2005, and 2007 respectively. Since then, she has been a visiting professor with Department of Mechanical Engineering at Koje College in Korea. Her main research interests include Computer Graphics and 3D Printing.

Young-Bong Kim, He received the B.S. in Computer Science from Seoul National University, Korea in 1987 and also received the M.S. and Ph.D. in Computer Science from Korea Advanced Institute of Science and Technology, Korea in 1989, 1994 respectively. Since then, he had worked as a researcher with Samsung Electronics until 1995. Since 1995, he has been a professor with Department of IT Convergence and Application Engineering at Pukyong National University in Korea. His main research interests include Computer Graphics and Animation.