CIT. Journal of Computing and Information Technology, Vol. 27, No. 4, December 2019, 31–46
doi: 10.20532/cit.2019.1004878


# Using Poisson Disk Sampling to Render Oil Particles at Sea

Vancuong Do[1, 2] and Hongxiang Ren[1]

[1]Key Laboratory of Marine Simulation and Control for Ministry of Communications, Dalian Maritime University, Dalian, China
[2]Faculty of Navigation, Vietnam Maritime University, Haiphong, Vietnam

Fluid simulation is one of the most complex tasks in three-dimensional simulation. Specifically, in the case of oil spills at sea, the oil film constantly interacts and is influenced by the environment, thus making its composition and properties change over time. In this paper, we tackle this problem by using both Lehr's spreading model and Hoult's drifting model to build the oil spill physical model. Unlike previous studies that only applied the Poisson disk algorithm to static and solid objects, we applied it in a three-dimensional space to divide the oil film into fluid particles. The track of oil particles under the influence of waves, wind, and currents is rendered by the Unity3D tool with C# programming language, which vividly and realistically simulates the collision of oil particles on the ocean scene with obstacles such as buoys and small islands. The result of this research can be used to predict oil spill direction, thus providing the solution to respond and minimize the damage caused by oil spills at sea. We also discuss some improvements to our model by using the Marching cube algorithm to render the Metaball model.

*ACM CCS (2012) Classification:* Computing methodologies → Computer graphics → Animation → Physical simulation

Computing methodologies → Computer graphics → Animation → Collision detection

Computing methodologies → Modeling and simulation → Simulation types and techniques → Real-time simulation

*Keywords*: fluid simulation, Poisson disk algorithm, oil spill, Unity3D

## 1. Introduction

As we know it, oil spill accidents often cause very serious and long-term consequences if they occur at sea, not only for the ocean ecosystem, economics and tourism, but also to the environment, especially if we do not respond timely and with preventive measures. The implementation of spills response is not always possible because of many reasons, partly due to the high cost of human and material resources, and partly due to weather conditions and sea areas inappropriate for drilling.

Nowadays, with the rapid development of science and technology, especially in the field of information technology, virtual reality simulations have greatly improved education and training in all areas of life. In our research we have applied virtual reality to simulate the development and transformation of an oil spill using Unity3D tool. The three-dimensional virtual reality simulation represents a visual and vivid sensation, which creates a real-world space to achieve a representation of highest efficiency. The model is a low-cost one to implement and does not require the use of equipment and staff on accident scene [1]. Also, we can change many variables, such as volume, location of oil, weather, waves, wind, and tidal flow, *etc.*, which are extremely important factors that cannot be controlled in the real world.

The rest of the paper is organized as follows. Related work done in this area is shortly reviewed in Section 2. We show the mechanical process of the oil film in Section 3. In Section 4, we provide an in-depth discussion of the Poisson disk sampling method and the way to apply it in the oil spill model. The results of our research are presented in Section 5, along with the related analysis. Section 6 brings the summary of our work, while some of its limitations are presented in Section 7, along with the directions for future work.

## 2. Related Work

Many authors have studied oil spill simulation models in both two-dimensional and three-dimensional scenarios as exemplified in related work. Hongxiang Ren *et al.* [2], [3] (2008, 2009) have applied the planar refraction map technique to render oil spilled in a three-dimensional scene, by using oil particles to calculate the track of oil slick with the assumption that each particle is of a cylinder shape, the height and base of a cylinder being equal. They also supposed that particles generation is so simple that they can be freely sampled within a circle. Feng Yu [4] (2010) used texture mapping to render the oil film on the sea. However, as the oil film is a unified block with a volume of 1000 $m^3$, it cannot realistically simulate an oil spill in 3D, and the initial oil particles are set randomly, too. Changjun Zou [5] (2018) used the NBFLIP method to simulate multi-phase fluids along with their collision with moving obstacles. His approach improves the computational efficiency of the simulation and can be used to simulate a multi-phase fluid, while having the particles generated using an organizational form method. Yang Zai-xing *et al.* [6] (2012) and Ran Jian, Ding Ying [7] (2012) used the Poisson disk distribution to simulate water in a river flow, however their model samples the incoherent part only, while the sampling area is too large hence reducing the computation speed. In our research, we apply the Poisson disk sampling technique in arbitrary dimensions to generate oil particles. This technique is based on the dart-throwing algorithm that generates uniformly distributed points. The simplest way to apply this technique was introduced by Robert Bridson [8] (2007), while in the recent decade many other researchers focused on how to innovate the Poisson distribution sampling method. Li-Yi Wei [9] (2008) improved the sampling speed by producing a parallel sampling method in a two-dimensional and three-dimensional arbitrary surface, which can run faster and is easy to execute. In their work, Sijia Liu *et al.* [10] (2016) investigated the impact of sampling strategies, where the Poisson disk sampling method was compared with other methods that include grid sampling, random sampling and jittered grid sampling. The Poisson disk method then provided a more accurate representation of the random region field. Bowers *et al.* [11] (2010) paper was also based on Wei's method of sampling on three-dimension arbitrary surfaces, however their method was shown to be highly efficient as it can be used to sample large scale objects. M. Ebeida *et al.* [12] introduced a maximal Poisson disk sampling to solve the problem of sampling uniform points, both maximal and unbiased in the sampling domain. Their method in turn used two phases, both based on classical dart-throwing. T. Jones [13] was also interested in a simple and efficient algorithm to generate Poisson disk sampling in a two-dimensional domain, and based his approach on the Voronoi diagram.

In this paper we apply the dart-throwing algorithm, since it not only generates points quickly and without the need for expensive hardware, something the previously mentioned methods have not met, but is also easy to implement. Poisson sampling makes the distribution of points more uniform, and effectively adjusts the sampling period so that the points achieve better distribution efficiency. The point density is not distributed according to Poisson's law, and is also neither too thick nor too thin. The area of the oil film is not too large, so the sampling rate is fast, hence satisfying real-time requirements. After sampling, the trajectory of oil slicks is shifted using Lehr and Hoult's numerical modeling [14], after which the collision detection of Rigidbody in Unity3D is used to implement the collision of oil particle with obstacles.

## 3. The Mechanical Process of Oil Film at Sea

A complete oil spill modeled simulation must consider all the processes of oil at sea, where these processes encompass the mechanical and physical transformation of the oil slick. The mechanical process includes all the changes in area, volume and position of the oil film from the original data, which is a spreading and drifting process. The latter process includes all the physical and chemical transformations which occur within the fluid itself, such as evaporation, emulsification, dissolution, photo-oxidation, bio-degradation, and sedimentation. However, within the scope of our study, we only investigated the mechanical process in the oil

spill simulation model, with the physical and biochemical processes being ignored.

### 3.1. Spreading

Spreading is not only important for response strategies, but it also affects the breaking processes. In the past, many researchers such as Brokker [15] (1964), Johansen [16] (1984), Elliot *et al.* [17] (1986) introduced oil spreading models. In order to describe the formula of oil spill rate, it is necessary to mention the classic formula of Fay [18] (1969). Fay divided surface spreading into three phases. The first phase is the gravity-inertial one, which is dominant in the first hour after the spill is released. This phase is the main surface spreading model for the simulation of oil spreading, and is applicable for the period of 1 hour to 1 week. The viscous-surface phase dominates when the oil film is broken into a smaller area of slick and persists after 1 week [19] [20]. However, this formula exhibits many limitations, such as describing the spread of oil on a calm sea surface, which does not include factors such as wind, currents, and waves. Since these are very important factors for the model accuracy, the oil film in Fay's formula is only a circle. Lehr has taken some tests of a spill in the Arabian Sea in 1982 and the result showed that the speed of oil spreading was underestimated.

When simulating the oil spreading, Lehr considered the effect of wind, where the oil film is an ellipse, with the major axis Q being the trend of wind direction and the minor axis R [21]. The area of the oil film can be hence predicted using the formula below:

$$A = (\pi / 4)QR \tag{1}$$

Lehr calculated the values of minor and major axes through experimentation, resulting in the equation (1) to be rewritten as follows:

$$A = 2270\left[\frac{\Delta\rho}{\rho_0}\right]^{\frac{2}{3}} V^{\frac{2}{3}} t^{\frac{1}{2}} + 40\left[\frac{\Delta\rho}{\rho_0}\right]^{\frac{1}{3}} V^{\frac{1}{3}} W^{\frac{4}{3}} t \tag{2}$$

$$A_{\max} = 10^5 V^{\frac{3}{3}} \tag{3}$$

where $A$ is the oil film area ($m^2$); $A_{max}$ is the maximum area the oil can spread ($m^2$); $V$ is the volume of oil (barrel); $W$ is wind rate (knot); $t$ is the time (min); $\rho_0$ is the density of oil; $\Delta\rho$ is the density difference between oil and water.

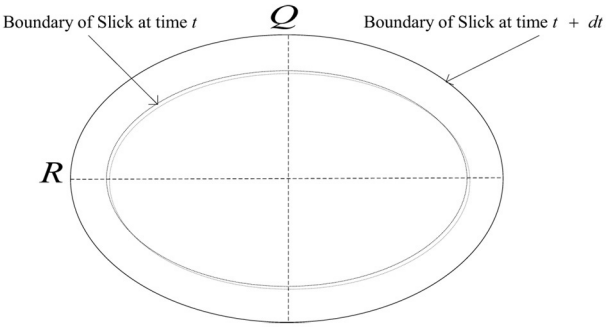



*Figure 1.* Spreading process in the Ellipse form.

### 3.2. Drifting and Diffusive

Drifting is the process that happens simultaneously with the spreading process after the oil is released. The factors that have a strong impact on the velocity and direction of oil drifting are wind, current and surface waves. Under these factors, our oil film is moving and breaks into many parts in other directions. Mackay and McAuliffe [22] showed that wind accounts for 3 % of the oil velocity. Figure 2 shows the relation of both wind and current components to the resulting movement of oil film direction. In his study [23] Fingas stated that if the oil slick is near an island or harbor and the wind velocity is not more than 10 km/h, the oil will drift at 100 % of the surface current and about 3 % of the wind speed. Our model is a discrete particle in which the particle is sampled under the Poisson disk algorithm. The movement of the oil film is now the movement of particles. The drifting velocity of a single oil particle can be written using the formula below:

$$\frac{dX_i}{dt} = V_{drift}(x_i, y_i, t) + V_{diff}(x_i, y_i, t) \tag{4}$$

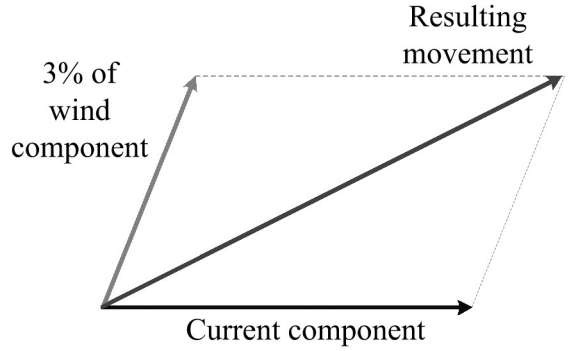
Here, $X_i(x_i, y_i)$ is the position of the $i$-th particle in the world space coordinate; $V_{diff}$ is the diffusion velocity; $t$ is the time (in seconds).

$$V_{drift}(x_i, y_i) = a_w V_w(x_i, y_i) + a_c V_c(x_i, y_i) \tag{5}$$

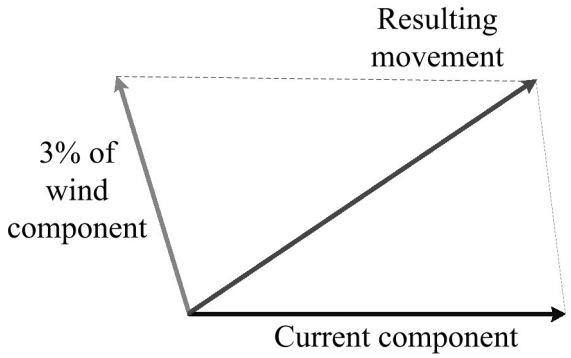
Here, $V_w$ is the wind velocity at 10 m above the water surface; $V_c$ is the current velocity; $a_w$ = 0.03 is the wind factor; $a_c$ = 1.0 is the current factor.

$$V_{diff}(x_i, y_i) = V_{drift} R_n e^{i\alpha_n} \quad (6)$$

Here, $R_n[-1, 1]$ is a random number; $\alpha_n[0, \pi]$ is a random angle; $V_{drift}$ is the drifting velocity.




*(a)* Down wind.




*(b)* Head wind.

*Figure 2.* Effect of wind and current direction on trajectory of oil spill.

## 4. Three-Dimensional Visualization of Oil Spill on the Sea

### 4.1. Poisson Disk Sampling Method

The definition of Poison disk sampling, which expresses the distribution of trees in a forest, was first introduced by the Swedish statistician Bertil Matérn in 1960 [24]. In computer graphics, this technique was first used by Dippé and Wold in 1985 for image anti-aliasing based on the dart throwing technique. Dart throwing [25], [26] is a classical technique for sampling Poisson disk distributions in which the points are randomly generated within a container, while they must satisfy the minimum distance $r$ to the other points.

In graphics applications, sampling is a technique that is very important, especially in rendering, imaging, and geometry processing. Poisson disk sampling is an algorithm that generates uniform random points on a field of $n$-dimensional space [27]. Samples are limited by the minimum distance criterion, which means that the distance between two points is greater than a certain threshold value, otherwise the candidates must be less than the maximum distance from some existing points, and each grid cell can hold one sample only. Because this algorithm is implemented in the scalar environment, the above minimum and maximum thresholds will establish two circles (in 2D), or two spheres (in 3D), with the corresponding radii of $r$ and $2r$, as is depicted in Figure 3.

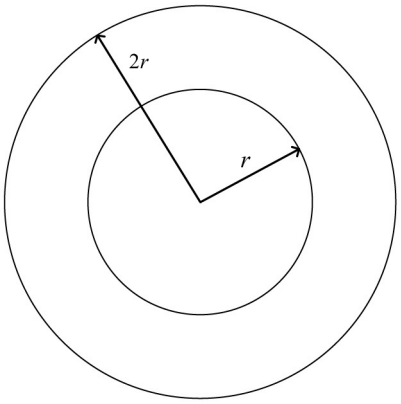



*Figure 3.* The threshold of candidates active.

Bridson introduced a method to generate points in the $R^n$ domain with minimum threshold $r$, and established the maximum candidates ($k$) to choose before canceling is $k = 30$. The details of the algorithm are as follows, and the algorithm itself is illustrated in Algorithm 1:

- First create an $n$-dimensional space grid that limits the sampling space and holds the entire sample in it.
- Divide the $n$-dimensional space into squares of size $\frac{r}{\sqrt{n}}$, so that each cell can hold one point only.
- Create an $n$-dimensional integer array containing the initialization points. When the

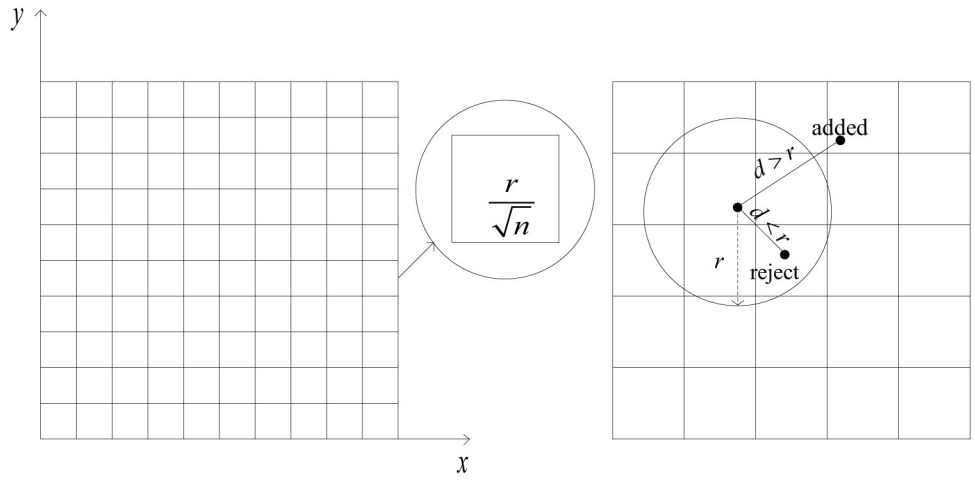



*Figure 4*. Background grid and distance condition of sampling.

array's element is not equal to 0, immediately generate $k$ points around it ($k$ = 30). For each candidate point, check the distance from it to the initial point, which has the index $x_i = 0$. If the candidate satisfies the condition $r \le d \le 2r$, add it to the active list and display it on the screen. If the candidate does not satisfy, then delete it and check the next point (up to the total of 30 candidates).

*Algorithm 1.* The Poisson disk sampling algorithm.

```
1.  // R^n is the sampling domain (n-dimensional space)
2.  // r is the minimum distance between samples
3.  // k is the maximum candidates that can be accepted, usually take k = 30
4.  // index is the value each grid is assigned to
5.  // d is the distance from candidate point to closest point
6.  function PoissonDiskSampling (R^n, r, k)
7.    setup grid dimensions
8.    cell size ← r/√n
9.    if grid [index] ≠ −1
10.     for all samples i from 0 to k do
11.       if r ≤ d ≤ 2r then
12.         candidate accepted
13.         add accepted candidate to the active list
14.       else candidate rejected
15.       end if
16.       i ← i + 1
17.     end for
18.   end if
19. end function
```

For the sake of simplicity, Figure 4 illustrates the Poisson disk algorithm for the 2-dimensional space. It is an example of a 2D background grid, and it has a clear depiction of the distance condition of sampling.

## 4.2. Dart Throwing

The dart-throwing algorithm was first introduced by Dippé and Wold, Cook and Mitchell [28] in the 1980s as a non-uniform sampling, in order to avoid aliasing problems in computer graphics. These distributions have blue noise properties. Blue noise means that the distance between two closest points follows a certain power law, under which high frequencies are more common. The less low-frequency noise generated, the better visually pleasing results for rendering, imaging and geometry processing ensue. However, although classic dart-throwing is easy and simple to achieve, it results in bad performance and slow speed. The algorithm is based on the approach that generates uniform distributions out of a set of dense points. It constantly checks the distance between the points until no points can be added any more. Wei (2008) [9] has solved many existing problems of classical dart throwing, proposing a parallel sampling method that can be used as a sequence of resolution levels to uniform grids in the dart-throwing process. This is the most popular Poisson disk sampling algorithm being used recently, which can be implemented on a Graphics Processing Unit (GPU). The method works by dividing the domain into sub-domains having

the same resolution level. For each resolution level, the cells are smartly classified into separate groups. Parallel dart-throwing will ensure that the cells in the same group can be sampled and that there are no risks of conflict between samples. Certainly, each cell can only contain one point. Each cell has $k$ dart throws to check the distance between the candidate points. If the candidate satisfies the distance condition it can be added to that cell, otherwise no sample is inserted into the cell. When all cells in the same group are sampled, the process checks the other groups having the same resolution level.

While the above research only generates uniform distributions of equal-sized samples, Cline *et al.* [29] presented an optimized dart-throwing algorithm to generate maximal Poisson disk sampling on three-dimension surfaces and developed the related algorithm to implement variable density of arbitrary samples without overlap. Optimized dart-throwing differs slightly from the classical one as follows. When darts are thrown using a varying sampling, both the minimum and maximum dart radiuses are taken. If the minimum dart radius is smaller than the radius of the sphere or ellipse, the dart is set with a random value within the minimum radius, and the minimum value between the maximum radius and the radius of sphere or ellipse from which the value is chosen, while any other way discards the process and checks the next one. We can observe the Cline's model process in Algorithm 2.

*Algorithm 2.* Dart throwing in a variable distribution.

1. // Setup: grid size, $k = 30$, $r$, $r_b$, rmax and $r_{\min}$, $n = 3$
2. **function** GeneratePoints(radius, sampleRegionsize, $k$)
3. setup grid dimension
4. cell size $\leftarrow \dfrac{r}{\sqrt{3}}$
5. **if** $r_b > r_{\min}$ **then**
6. $r$ = Random.Range($r_{\min}$, min($r_{\max}$, $r_b$))
7. **else** discard the dart
8. expands to the nearby spheres
9. **end** function

Figure 5 shows point generation on the Unity3D platform using Poisson disk sampling with a variable size of spheres, as implemented by us.

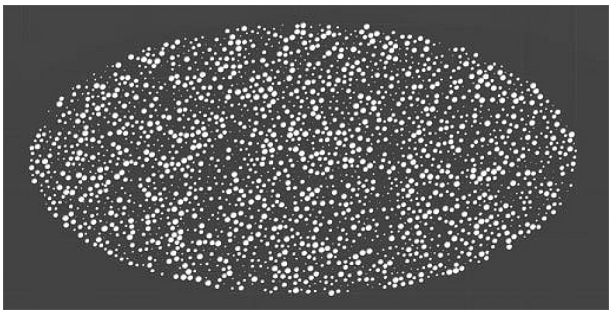


*Figure 5*. Poisson disk sampling, with a variable distribution.

## 4.3. Application of Poisson Disk Algorithm on the Oil Model

Within the present research, we assume that the oil film will spread under the effect of gravity, inertial, viscous and surface tensions. This makes the oil reach a constant level by spreading horizontally, until the moment when the oil surface layer is thin enough, and when Fay's spreading process is stopped. Lehr estimates the maximum area of such spill in Eq. (3). When Fay's spreading process has ceased, the oil film still spreads, but in this case, the nature of spreading is different. Namely, the oil film is dispersed into smaller parts and is separated under the action of waves and surface currents. In order to implement this process, we have used the Poisson disk sampling to divide the maximum oil area (of ellipse shape) into particles. Furthermore, we consider the particle system to consist of a thousand spheres, which is a random sampling within the area of an ellipse. In Figure 6, $O(x, y, z)$ is the origin, $Oxz$ is the oil film plane, the y axis is the peak of sea level, and we consider the y value to be initially equal to 0. Mathematically, the way how to check whether a point is inside or outside of an ellipse can be determined using the inequality:

$$\frac{(x-h)^2}{a^2} + \frac{(y-k)^2}{b^2} \leq 1 \qquad (7)$$

where $(h, k)$ is the center of the ellipse, while $a$ and $b$ are the semi-major and semi-minor of the ellipse, respectively.

Assuming that the sampling domain is scalar in the $Oxz$ plane, we will accept the points inside the ellipse, while others that are not on the

boundary or inside the ellipse will be removed. To implement this, we apply the following inequality:

$$\frac{\left(x_i - \frac{a}{2}\right)^2}{\left(\frac{a}{2}\right)^2} + \frac{\left(z_i - \frac{b}{2}\right)^2}{\left(\frac{b}{2}\right)^2} \leq 1 \qquad (8)$$

where $x_i$, $z_i$ are coordinates of the $i$-th point on the $x$ and $z$ axes, respectively, and $a/2$ and $b/2$ are the semi-major and semi-minor axes, respectively.

We express the application of Poisson disk sampling on our oil film model in Algorithm 3.

*Algorithm 3.* Poisson disk algorithm used on the oil model.

```
1.  // Setup: grid size, k = 30, radius, n = 3
2.  // Index: the value each point is assigned
3.  // ValidPoint: a function to check whether a point
    is valid or not
4.  // d: distance from ValidPoint to existing point
5.  function GeneratePoints(radius,
    sampleRegionsize, k)
6.      setup grid dimensions
7.      cell size ← r/√3
8.      if grid has element then
9.          for all samples i from 0 to k do
10.             if ValidPoint is true then
11.                 candidate accepted
12.                 add candidate to the active list
13.             else candidates rejected
14.             end if
15.             i ← i + 1
16.         end for
17.     end if
18. end function
19. function ValidPoint(points, radius, k,
    sampleRegionsize, Gridsize)
20.     if points is in the ellipse then
21.         for all points in the valid region do
22.             if points index ≠ −1 then
23.                 check r ≤ d ≤ 2r
24.                 return true
25.             end if
26.         end for
27.     end if
28. end function
```

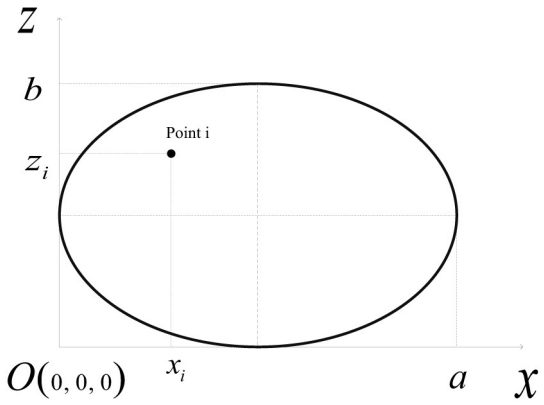
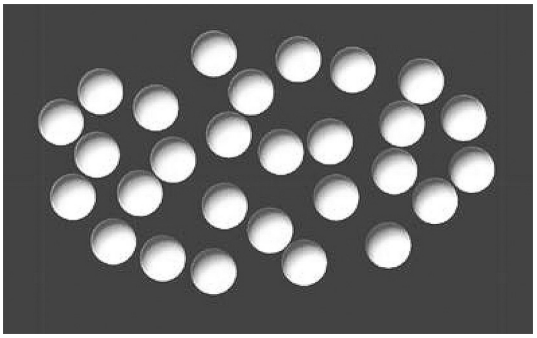
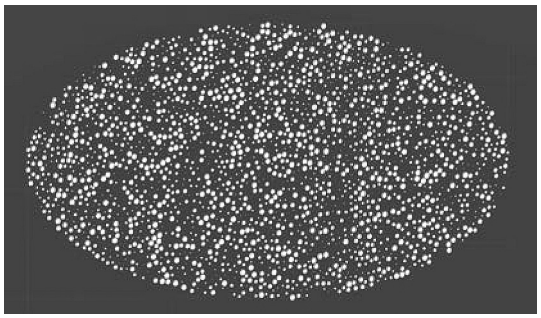



*Figure 6.* Sampling region in 3D space.

## 4.4. Quantity and Scale of Particles



*(a)* 28 particles model.



*(b)* 5000 particles model.

*Figure 7.* Quantity and scale of particles in the model.

The quantity of particles is a very important factor in the oil spill model. The larger the number of the particles, the more flexible and realistic is the oil's movement. Conversely, if the number of particles is too small, the model cannot exactly simulate the characteristics of oil leakage at sea. Again, if the number is too large, it could require a higher quality hardware configuration, but would in turn significantly

reduce the computation speed. Therefore, we need to base the quantity of particles to be sampled in our model on both RAM capacity and CPU/GPU computation speed of the computer used in the simulation. We have performed the sampling for two cases, as shown in Figure 7: on the left side (Figure 7a), the simulation was performed for a small number of equal spheres, while on the right side (Figure 7b), the optimized dart-throwing algorithm of Cline is used to sample about 5000 spheres with different diameters.

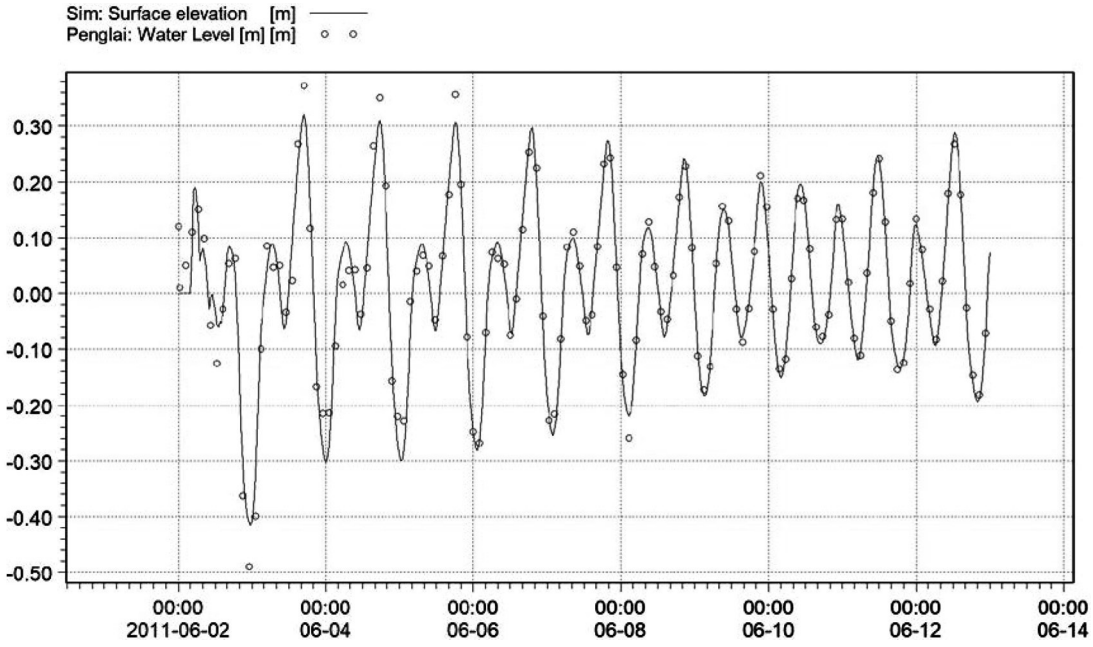
### 4.5. Tidal Calculation

In Section 2 above, we have noted the significance of drifting and diffusion to the oil spill model. Current and wave direction are two factors that greatly affect the accuracy of the oil spill model. Within the scope of this paper, we use the Mike 21 model, a part of the Mike Zero software package, to calculate the tidal information of the sea area that we assume has been subject to an oil spill accident. Mike 21 [30], [31], [32] is an application of the hydrodynamic model used to calculate the mass and momentum in the vertical axis, following the law of conservation of mass and momentum. The hydrodynamic module is a combination of the two-dimensional shallow water equations, *i.e.* the depth-integrated incompressible Reynolds averaged Navier-Stokes equations:

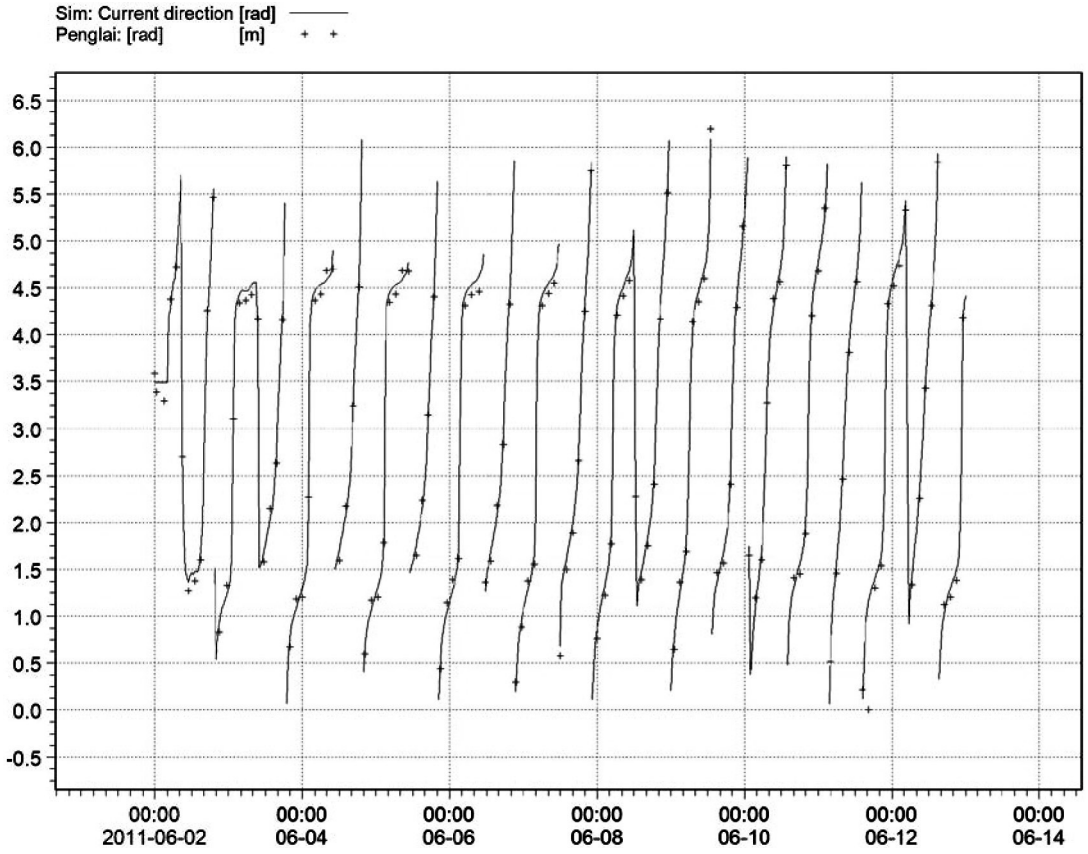
$$\frac{\partial \zeta}{dt} + \frac{\partial p}{dx} + \frac{\partial q}{dy} = \frac{\partial d}{dt} \tag{9}$$

where $\zeta$ (m/s) is the variation of surface elevation over time $t$ (sec); $p$ and $q$ are flux densities in $x$ and $y$ directions ($m^3/s/m$); $d$ is the time varying water depth (m).

We used Mike Zero to generate a mesh, and interpolate boundary conditions, and then Mike 21 Flow Model to shape tidal data such as current speed, current direction, wave direction, surface elevation, and the like. In this paper, we chose the Bohai Bay sea area to simulate the tidal wave at a proper moment, afterwards comparing our tidal model with the 2011 observation data of the ZhiFuDao observation station located close to the Penglai oil rig position. The observation data were collected from the website of the National Marine Science Data Center, National Science & Technology Resource Sharing Service Platform of China [33]. The comparison showed a high similarity between the two, as visible in Figure 8, where water level and current direction are depicted (the solid line denotes the simulation results, while the dotted one the observation results). Figure 9 shows the current direction of the Bohai Sea at the ebb tide and high tide.

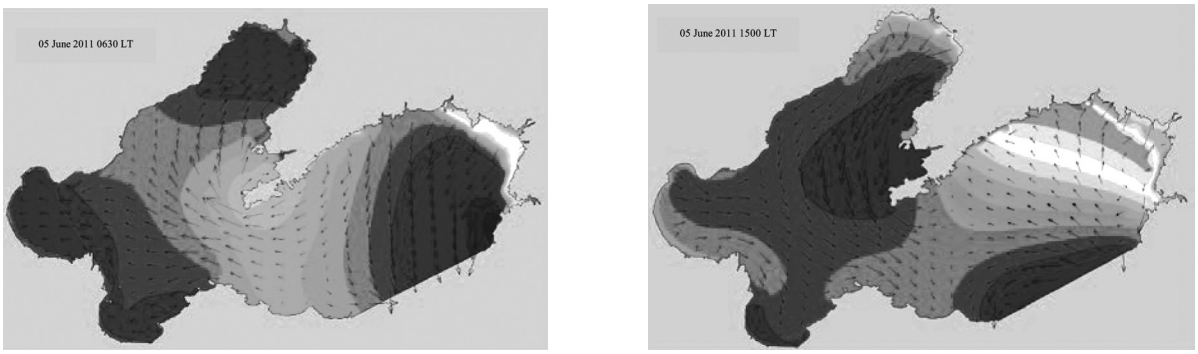



*(a)* Water level




*(b)* Current direction

*Figure 8.* Comparison of simulation and observation data.




*(a)* Hight tide *(b)* Ebb tide

*Figure 9*. Bohai Bay current direction at high and ebb tide.

## 5. Real-Time Simulation and Visualization of Oil Spill

### 5.1. Implementation Process

Our research mainly relates to an instantaneous oil spill process, and considers the two characteristics of spreading and drifting. In previous research many authors focused on both emulsion and evaporation processes besides calculating spread and drift, while our model is an emergency oil spill rescue model. Since the life cycle of oil at sea is short, it is difficult to take into account the evaporation and emulsion processes, thus actions are quickly carried out after an instantaneous oil spill occurs. In this respect, our model is perfectly suitable.

Our oil spills model is given as a step-by-step process in Figure 10. Firstly, we set up the initial data of an oil spill such as oil type, volume, density, oil temperature, *etc.* as well as the weather condition considering the wind data at the level of 10 meters above water surface. Mike Zero and Mike 21 Flow Model FM was used to calculate the tidal data. These data were subsequently used as inputs to the oil spreading and oil drifting model to calculate the new shape and position of the oil film based on Lehr's research. After reaching the maximum area, we immediately apply the Poisson disk algorithm to render oil film into particles, and recalculate the oil position again. Both movement and diffusion of oil particles as affected by waves, wind, and current, are all rendered within a single platform (*i.e.* Unity3D) thus producing a nice oil spill simulator canvas.

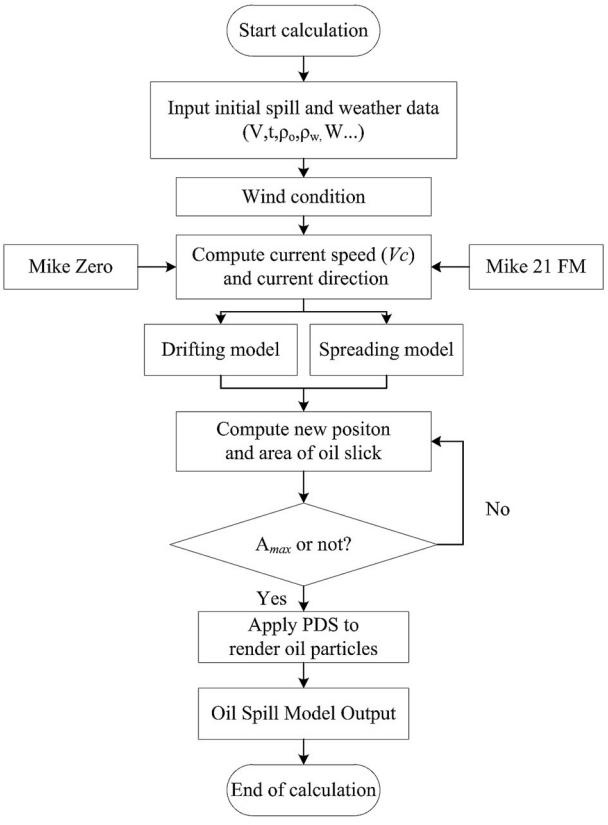



*Figure 10*. Flowchart of oil spill simulation process.

## 5.2. Visualization of Oil Spill Model

Visualization of the oil spill model has been performed on a portable computer with the following characteristics:

- chipset: Intel (R) Core(TM) i7-6700 CPU @ 3.40 GHz;
- graphic card: NVIDIA Quadro K620;
- memory: RAM DDR4 8GB;
- CPU: 4 cores and 8 threads.

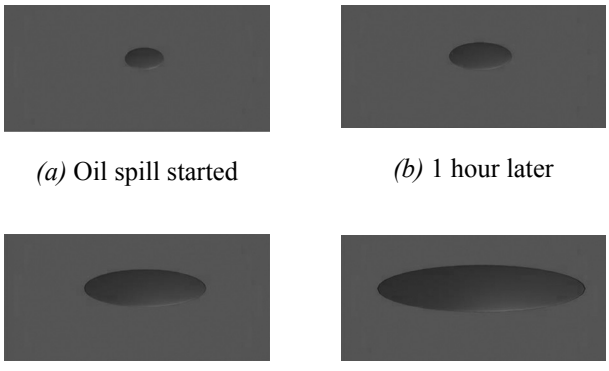
### 5.2.1. Oil Spill Model According to Lehr Period

To build our product capable of three-dimensional modelling, we used the Microsoft Visual Studio 2013 development tools with C# language and Unity3D development platform.

We assumed the following input data:

- simulation time: 04 Jul 2011;
- spill position: near the point has coordinated: 120.10E, 38.40N;
- oil volume release: 1000 $m^3$;
- oil density: $\rho_0 = 806$ kg/$m^3$;
- water density: $\rho_w = 1025$ kg/$m^3$;
- wind speed at 10 m above water surface: $W_{10} = 0.5$ m/s.

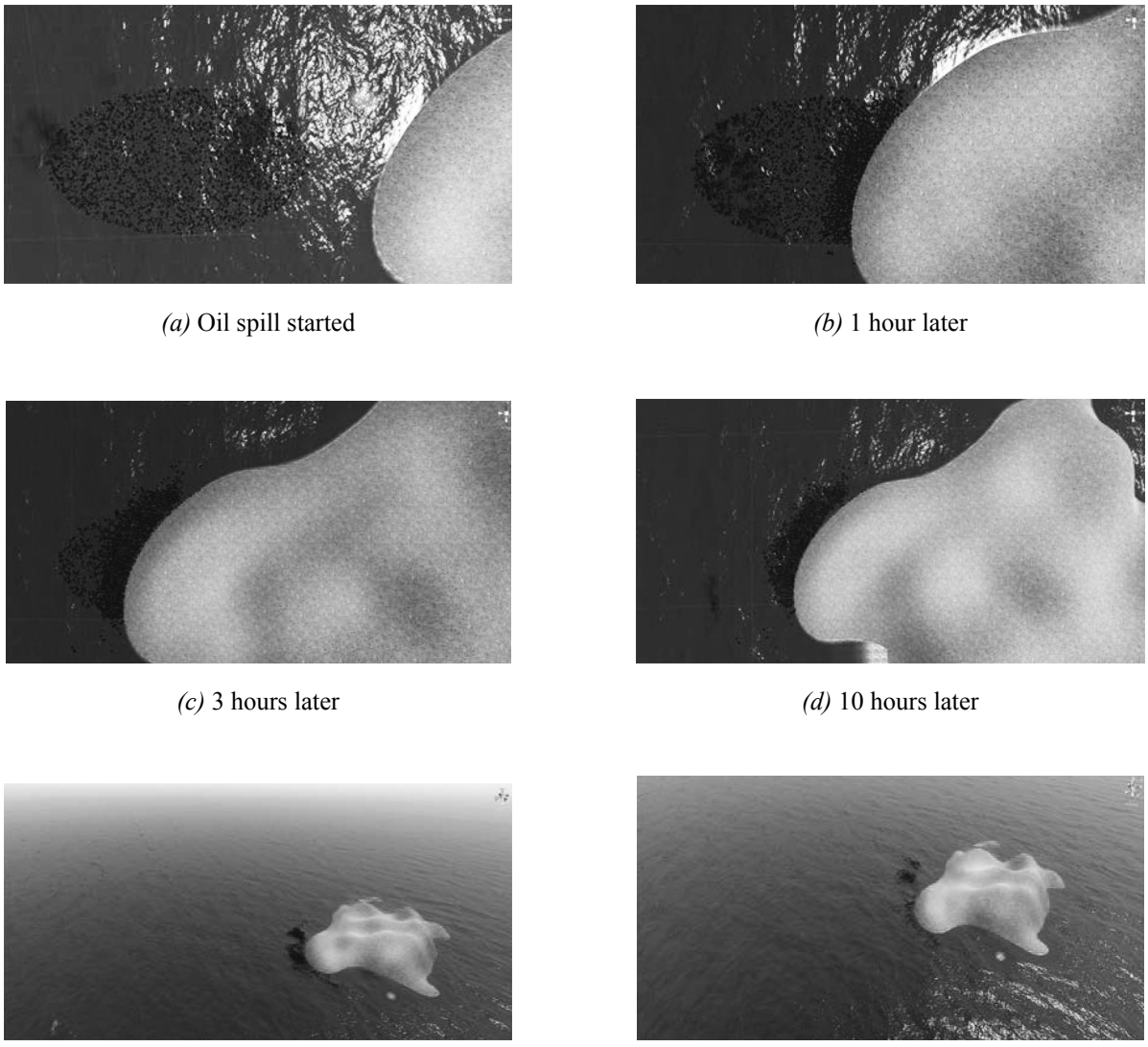
Figure 11 shows the results of a Unity3D simulation of the initial process of an oil spill, which is based on Lehr's model. In this process, the oil film is only a development of an ellipse, whose area of can be calculated after equations (1) and (2). The individual figures show the oil spill spread at the inception, 1 hour later, 3 hours later and when reaching the maximum area.



*(a)* Oil spill started *(b)* 1 hour later

*(c)* 3 hours later *(d)* Maximum area

*Figure 11.* The process of oil spill visualization.



*(a)* Oil spill started

*(b)* 1 hour later

*(c)* 3 hours later

*(d)* 10 hours later

*(e)* 24 hours later

*(f)* 48 hours later

*Figure 12.* Visualization of oil spill meeting obstacle objects.

#### 5.2.2. Oil Spill Model when Poisson Disk Sampling Was Applied

When the oil film reaches the maximum area, we divide the oil shape into thousands of particles basing on the Poisson disk sampling method, in which all of these oil particles have different sizes and diameters hence ensuring that our model can satisfy all the requirements for an adequate visualization. We consider two cases of oil spill at sea:

(i) moving oil particles, eventually reaching an island;

(ii) oil particles continuously spreading and drifting under the effect of environmental factors.

The former case is illustrated in Figure 12, while the latter one in Figure 13. We used the Poisson disk sampling algorithm to generate about 2500 oil particles, then render all of them on an ocean background; and calculated the buoyancy, color and other characteristics of oil particles to make our model in the best performance way. The oil film was continuously updated, with results at characteristic moments presented in Figures 12 and 13.

Specifically, Figure 13d can be compared to new research of Lehr on the extent of oil spreading in 2017. At the beginning of oil release, the slick was still thick and could not dampen the wave and surface tension, so the windrow might not be that apparent. When

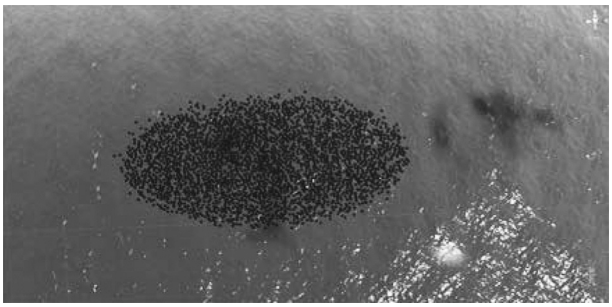
the slick was thin enough, the wave and wind factors dampened, and the windrow could be easily observed [34]. We also made a comparison with real-life situations of the Deep Water Horizon oil spill in July 2010. The similarities between our model and the latter oil spill are shown in Figure 14.

Presently there are very few 3D visualization researches of oil spill model at sea. Ren Hong Xiang [2] made a preliminary attempt to 3D visualization of oil spills on a navigation simulator, using a simple and practical method to calculate tidal and wind factors. He also applied the oil particle model to show how the oil slick was broken up into smaller parts on the water surface, however, he considered the particles to be small cylinders of the same size. Within the spreading model he used Fay's equation to calculate the oil spill area at interval time, resulting in the oil slick being represented as a circle on the water surface. A number of limitations were additionally revealed when using Fay's formula, making thus the oil spill prediction model to be of low accuracy. Ren's 3D visualization oil spill model is shown in Figure 15.

Yu Feng also researched 3D visualization oil spill models at sea, using Fay's formula for the spreading model, but not using the particle model at all. This however resulted in her model not satisfying the breaking up of the oil slick into smaller parts, which can be seen in Figure 16.

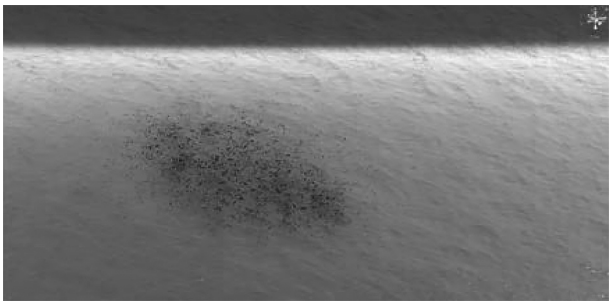
## 6. Conclusion

Fluid simulation is a difficult part within the area of computer graphics but has many applications in real life. In the research described in this paper we used the Poisson disk sampling method to solve the most difficult part of fluid simulation, namely particles sampling. In our model, the spreading and drifting of oil spills were based on Lehr's equations producing a good result. Mike Zero was used to solve the tide problem, while the Mike 21 Flow Model was used to compute current speed, current direction, and other tidal data. The other important task in simulations was to render and visualize the results of calculations for a suitable presentation to the user. Here, we chose Unity3D, a known professional game development platform. The developed model can be used for training and as a part of exercises within marine rescue simulator systems.

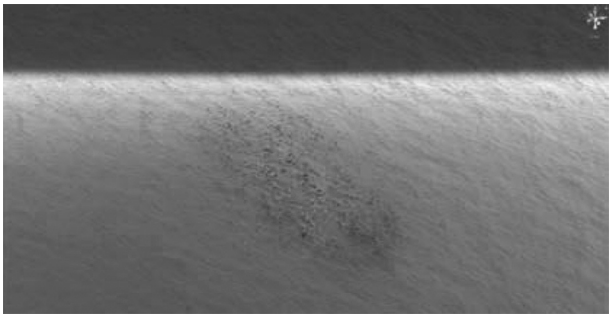


*(a)* Oil spill started

*(b)* 3 hours later

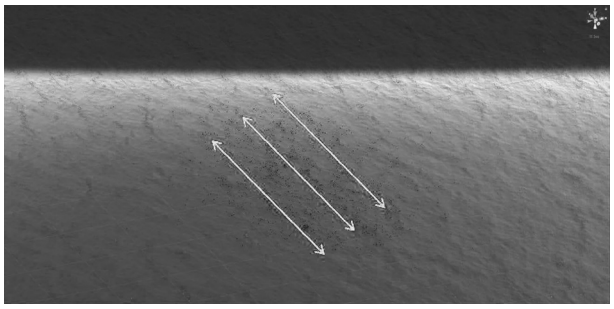


*(c)* 10 hours later



*(d)* 24 hours later

*Figure 13.* Oil spill visualization meeting no obstacle object.



*(a)* Windrow in our model

*(b)* Windrows resulting in Deep Water Horizon oil spill

*Figure 14.* Comparison with the real-life situation.

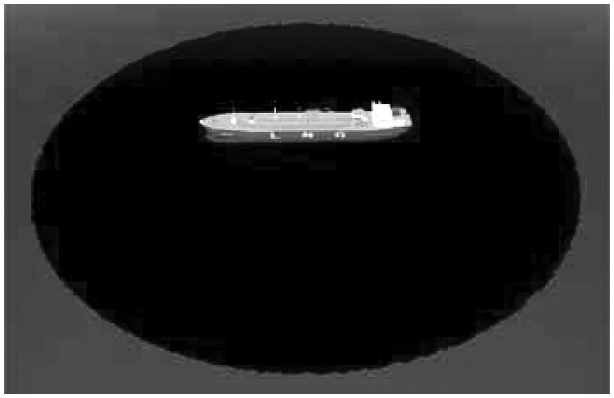


*Figure 15.* Ren's 3D oil spill model.

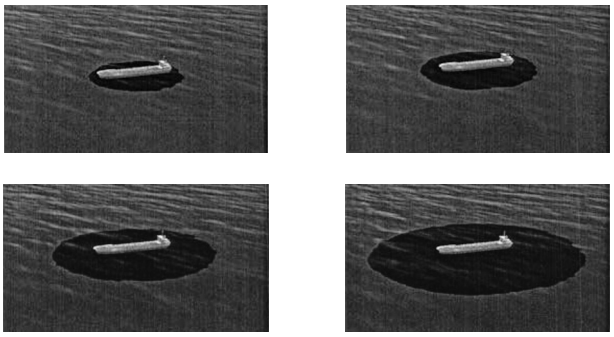


*Figure 16.* Yu's 3D oil spill model.

## 7. Limitations and Future Work

Although the Poisson disk sampling algorithm made a breakthrough in describing the characteristics of oil spills at sea, thus providing for an easy rendering of its movement through the application of mathematical formulas to each oil particle, there are still limitations in our model. The oil shape is only a cluster of a thousand particles which have not completely satisfied the fluid simulation. In future work, we plan to use the Marching cube algorithm to complement the present work. Marching cube is a computer graphics algorithm for extracting a polygonal mesh of an iso-surface from a three-dimensional scalar field which can be called a voxel, developed by Lorensen and Cline and originally presented at the 1987 SIGGRAPH Conference [35]. The algorithm has been applied to multiple fields, including video games, medical visualizations like Computer Tomography (CT) and Magnetic Resonance Imaging (MRI) scan data images, and special effects or 3D modeling with what is usually called metaballs or other metasurfaces. An analogous two-dimensional method is called the Marching squares algorithm.

The Marching cube algorithm uses an iso-surface to define which vertex is inside or outside the cross-plane, after which a surface through these cut-points can be drawn [36]. This technique allows the conversion of a non-polygonal 3D model (such as the one represented by voxels) to a polygonal one. It works by examining cube-shaped chunks of the model and representing each chunk with a certain pre-defined arrangement of polygons depending on which corners of the cube are solid. Metaballs provide a non-polygonal way of defining 3D models mathematically as a set of 3D points (the "metaballs") that sort of glob together as if they were made of magnetic goo. Metaball models tend to do well at curvy surfaces and not very well at sharp, angular, or flat ones. The Marching cube algorithm can be used to convert a metaball model into a polygonal one, and subsequently easily render it by using GPUs. Our approach is to apply the Marching cube algorithm to represent metaballs, where each metaball represents an oil particle, eventually achieving the effect of the oil film on the sea. This is shown in Figure 17, where the simulation is performed by

the Unity3D game engine. This approach can be used for fluid simulations such as water droplets or oil spills.

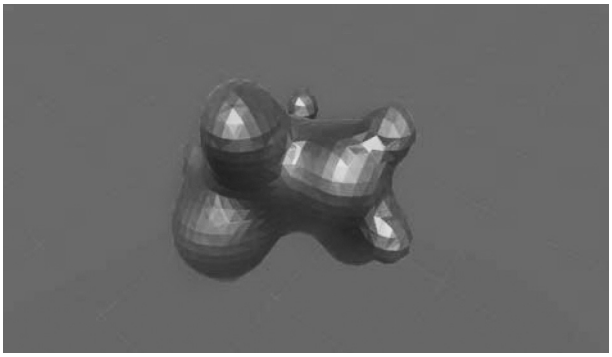


*Figure 17.* An application of the Marching cube algorithm to render a metaball.

## Acknowledgment

This work was supported and sponsored by the National High Technology Research and Development Program of China (863 Program No. 2015AA016404) and the Fundamental Research Funds for the Central Universities (No. 3132016310). We also acknowledge the help in data support from the National Marine Science Data Center, National Science & Technology Resource Sharing Service Platform of China.

## References

[1] X. Chen *et al.*, "Research of 3D Oil Spill Response Drill System", *Aquatic Procedia*, vol. 3, pp. 15–20, 2015. http://dx.doi.org/10.1016/j.aqpro.2015.02.222

[2] H. Ren *et al.*, "3D Real-Time Rendering of Continuous Oil Spill on Simulated Ocean", *Journal of System Simulation*, vol. 20, no. 19, pp. 5117–5120, 2008. http://dx.doi.org/10.16182/j.cnki.joss.2008.19.047

[3] H. Ren *et al.*, "3D Visualization of Oil Spill in Marine Simulator", *Journal of System Simulation*, vol. 21, no. 1, pp. 161–165, 2009. (in Chinese) http://dx.doi.org/10.16182/j.cnki.joss.2009.01.041

[4] Y. Feng and Y. Yong, "Real-Time Visualization and Simulation of Oil Spill Based on Physical Model", *Journal of System Simulation*, vol. 20, pp. 313–315, 2008. (in Chinese) http://dx.doi.org/10.16182/j.cnki.joss.2008.s1.041

[5] Z. Changjun and Y. Yong, "Fluid Simulation Based on Narrow Band FLIP Method", *Journal of Computer-Aided Design and Computer Graphics*, vol. 30, no. 4, pp. 577–583, 2018. (in Chinese) http://dx.doi.org/10.3724/SP.J.1089.2018.16458

[6] Y. Zaixing *et al.*, "Simulation of Flowing Water Based on Flow Field", *Computer Engineering and Design*, vol. 33, no. 6, pp. 2392–2397, 2012. (in Chinese) http://dx.doi.org/10.16208/j.issn1000-7024.2012.06.001

[7] R. Jian and D. Ying, "River Simulation Based on Poisson Disk Distribution", *Journal of Beijing University of Aeronautics and Astronautics*, vol. 38, no. 12, pp. 1649–1652, 2012. (in Chinese) http://dx.doi.org/10.13700/j.bh.1001-5965.2012.12.005

[8] R. Bridson, "Fast Poisson Disk Sampling in Arbitrary Dimensions", in *ACM SIGGRAPH sketchs*, vol. 22, p. 3, 2007. http://dx.doi.org/10.1145/1278780.1278807

[9] L.-Y. Wei, "Parallel Poisson Disk Sampling", *ACM Transactions on Graphics*, vol. 27, no. 3, pp. 1–9, 2008. http://dx.doi.org/10.1145/1360612.1360619

[10] S. Liu *et al.*, "Sensor Placement for Field Estimation via Poisson Disk Sampling", in *Proc. of the IEEE Global Conference on Signal and Information Processing, GlobalSIP*, 2016, pp. 520–524. http://dx.doi.org/10.1109/GlobalSIP.2016.7905896

[11] J. Bowers *et al.*, "Parallel Poisson Disk Sampling with Spectrum Analysis on Surfaces", *ACM Transactions on Graphics*, vol. 29, no. 166, pp. 1–10, 2010. http://dx.doi.org/10.1145/1866158.1866188

[12] M. S. Ebeida *et al.*, "Efficient Maximal Poisson Disk Sampling", *ACM Transaction on Graphics*, vol. 30, no. 4, pp. 49:1–49:12, 2011. http://dx.doi.org/10.1145/2010324.1964944

[13] T. R. Jones, "Efficient Generation of Poisson Disk Sampling Patterns", *Journal of Graphics Tools*, vol. 11, no. 2, pp. 27–36, 2005. http://dx.doi.org/10.1080/2151237X.2006.10129217

[14] D. P. Hoult, "Oil Spreading on the Sea", *Annu. Rev. of Fluid Mech.*, pp. 341–368, 1972. http://dx.doi.org/10.1146/annurev.fl.04.010172.002013

[15] P. C. Brokker, "Spreading and Evaporation of Petroleum Products on Water", in *Proc. of the 9th International Habour Conference*, Antwerp, 1964.

[16] O. Johansen, "The Halten Bank Experiment – Observations and Model Studies of Drift and Fate of Oil in the Marine Environment", in *Proc. of the 11th Arctic Marine Oil Spill Program Technology Seminar*, Ottawa, Ontario: Environment Canada, pp. 18–36, 1984.

[17] A. Elliot *et al.*, "Shear Diffusion and the Spreading of Oil Slicks", *Marine Pollution Bulletin*, vol. 17, no. 7, pp. 308–313, 1986. http://dx.doi.org/10.1016/0025-326X(86)90216-X

[18] J. A. Fay, "The Spread of Oil Slicks on a Calm Sea", in D. Hoult (Ed.), *Ocean Technology*, Plenum Press, pp. 53–63, 1969. http://dx.doi.org/10.1007/978-1-4684-9019-0_5

[19] A. H. Al-Rabeh *et al.*, "A Stochastic Simulation Model of Oil Spill Fate and Transport", *Applied Mathematical Modelling*, vol. 13, no. 6, pp. 322–329, 1989. http://dx.doi.org/10.1016/0307-904x(89)90134-0

[20] P. Sebastiao and C. G. Soares, "Modeling the Fate of Oil Spills at Sea", *Spill Science and Technology Bulletin*, vol. 2, no. 2/3, pp. 121–131, 1995. http://dx.doi.org/10.1016/S1353-2561(96)00009-6

[21] W. J. Lehr *et al.*, "A New Technique to Estimate Initial Spill Size Using a Modified Fay-Type Spreading Formula", *Marine Pollution Bulletin*, vol. 15, no. 9, pp. 326–329, 1984. http://dx.doi.org/10.1016/0025-326x(84)90488-0

[22] D. Mackay and C. D. McAuliffe, "Fate of Hydrocarbons Discharge at Sea", *Oil Chem. Pollut.*, vol. 5, no. 1, pp. 1–20, 1989. http://dx.doi.org/10.1016/S0269-8579(89)80002-4

[23] M. Fingas, "Introduction to Spill Modeling (Chapter 8)", *Oil Spill Science and Technology*, pp. 187–200, 2011. http://dx.doi.org/10.1016/B978-1-85617-943-0.10008-5

[24] M. N. Gamito and S. C. Maddock, "Accurate Multidimensional Poisson Disk Sampling", *ACM Transactions on Graphics*, vol. 29, no. 1, Article 8, pp. 1–19, 2009. http://dx.doi.org/10.1145/1640443.1640451

[25] R. Anirudh *et al.*, "Poisson Disk Sampling on the Grassmannnian: Applications in Subspace Optimization", in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Hawaii, 2017, pp. 690–698. http://dx.doi.org/10.1109/CVPRW.2017.98

[26] Y. Xiang *et al.*, "An Intrinsic Algorithm for Parallel Poisson Disk Sampling on Arbitrary Surfaces", *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 9, pp. 1425–1437, 2013. http://dx.doi.org/10.1109/TVCG.2013.63

[27] A. Lagae and P. Dutré, "A Comparison of Methods for Generating Poisson Disk Distributions", *Computer Graphics Forum*, vol. 27, no. 1, pp. 114–129, 2008. http://dx.doi.org/10.1111/j.1467-8659.2007.01100.x

[28] R. L. Cook, "Stochastic Sampling in Computer Graphics", *ACM Transaction on Graphics*, vol. 5, no.1, pp. 51–72, 1986. http://dx.doi.org/10.1145/7529.8927

[29] D. Cline *et al.*, "Dart Throwing on Surfaces", *Computer Graphics Forum*, vol. 28, no. 4, pp. 1217–1226, 2009. http://dx.doi.org/10.1111/j.1467-8659.2009.01499.x

[30] "Mike21 Flow Model FM, Hydrodynamic Module, User Guide", DHI 2017.

[31] "Mike21 Flow Model & Mike21 Flood Screening Tool, Hydrodynamic Module, Scientific Documentation", DHI 2017.

[32] "Mike21 Tidal Analysis and Prediction Module, Scientific documentation", DHI 2017.

[33] National Marine Science Data Sharing Service Platform. http://mds.nmdis.org.cn/

[34] D. S. Beatty and W. J. Lehr, "Extended Oil Spill Spreading with Langmuir Circulation", *Marine Pollution Bulletin*, vol. 122, pp. 226–335, 2017. http://dx.doi.org/10.1016/j.marpolbul.2017.06.047

[35] W. Lorensen and H. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm", in *Proc. of the SIGGRAPH'87*, 1987, pp. 163–169. http://dx.doi.org/ 10.1145/37402.37422

[36] B. Kenwright, "Metaballs & Marching Cubes: Blobby Objects and Isosurfaces", *Technical Article*, pp. 1–7, 2014.

*Received:* September 2019
*Revised:* March 2020
*Accepted:* April 2020

*Contact addresses*:
Vancuong Do*
Key Laboratory of Marine Simulation and Control for Ministry of Communications
Dalian Maritime University
Dalian
China
e-mail: dovancuong@vimaru.edu.vn
*Corresponding author

Hongxiang Ren
Key Laboratory of Marine Simulation and Control for Ministry of Communications
Dalian Maritime University
Dalian
China
e-mail: dmu_rhx@163.com

Vancuong Do was born in August 1988 in Hanam province, Vietnam. He received his Bachelor's degree in the field of Navigation, from the Navigation Faculty, Vietnam Maritime University, in 2011. He completed his Master of Science in Navigation at the Navigation Faculty, Vietnam Maritime University in 2014. He has been a PhD candidate in Navigation Science and Technology major, Dalian Maritime University, China, since 2016. In 2011, he joined the Navigation Faculty, Vietnam Maritime University, as a teaching assistant and then a lecturer. His research interests are computer graphics, navigation simulation, and virtual reality. He presented some papers at the IEEE conference which considered the virtual reality and tidal simulation at sea. In recent years, he has focused on fluid simulation and collaborated with the professors and researchers at the Key Laboratory of Marine Simulation and Control for Ministry of Communications, Dalian Maritime University in the field of navigation science, virtual reality and computer graphics.

Hongxiang Ren was born in September 1974 in P. R. China. In 2009, he received his PhD degree in the field of Transportation Engineering (Scene Simulation) from the Key Laboratory of Marine Simulation and Control for Ministry of Communications, Dalian Maritime University, China, Master's degree in Machinery and CAD, Harbin University of Science and Technology, China, in 1998, and Bachelor's degree in Mechanics, Harbin University of Science and Technology, China, in 1995. His research interests are computer graphics, virtual reality, and scene simulation. He has worked at Navigation School, Dalian Maritime University as a lecturer since 2000 and professor since 2010. Prof. Hongxiang Ren has received many awards from the Government of the People's Republic of China and Liaoning Province. In addition, he became the visiting scholar at the University of Tennessee, America, in 2015. He has published many SCI and EI compendex papers related to fluid simulation, navigation simulation system, and computer graphics.