

Куров Андрей Владимирович,
канд. техн. наук, доцент МГТУ им. Н.Э.Баумана
E-mail: avkur7@mail.ru

Завалин Андрей Александрович,
студент первого курса магистратуры МГТУ им. Н.Э.Баумана
г.Москва, РФ
E-mail: aazavalin@yandex.ru

ВИЗУАЛИЗАЦИЯ ВОДНОЙ ПОВЕРХНОСТИ НА ОСНОВЕ СПЕКТРАЛЬНОГО МЕТОДА

Аннотация

Дан краткий обзор существующих подходов к решению задачи визуализации поверхности воды. Детально рассмотрен спектральный метод, позволяющий генерировать модель поверхности воды в реальном времени. Содержится описание метода визуализации заострённых волн, позволяющего увеличить реалистичность получаемой при помощи спектрального подхода модели. Разработана программа визуализации водной поверхности при помощи спектрального подхода в реальном времени, использующая технологию OpenGL Compute Shaders, позволившую распараллелить алгоритм быстрого преобразования Фурье на GPU, добившись тем самым увеличения производительности.

Ключевые слова

Визуализация, водная поверхность, спектральный подход, генерация поверхности воды, быстрое преобразование Фурье, OpenGL Compute Shaders.

Одной из задач современной компьютерной графики является создание реалистических изображений различных объектов и явлений окружающей среды. В последнее время внимание многих исследователей привлекает моделирование водной поверхности. Это объясняется тем, что создание реалистических изображений поверхностей различных водоемов (озер, рек, океанов) требует выполнения моделирования водной поверхности с учетом возникающих, в силу разных причин, на ее поверхности волн.

Поскольку моделирование может выполняться на технических средствах, существенным образом различающихся своими техническими характеристиками (например, персональные компьютеры и мощные кластеры), то возникает потребность в создании и применении разных моделей. Известные модели постоянно подвергаются изменениям: одни упрощаются или оптимизируются и становятся доступными для реализации на менее мощных вычислительных машинах, другие совершенствуются и позволяют получить наиболее реалистичное изображение на мощных ЭВМ.

Модели визуализации воды в зависимости от используемого подхода к решению проблемы могут быть разделены на две группы: реализующие физический или процедурный (параметрический/спектральный) подход. В первом случае в моделях задействованы методы вычислительной гидродинамики, использующие уравнения, описывающие рассматриваемые физические процессы. Несмотря на то, что эти методы применимы в моделировании не только воды, но и любой жидкости, а также способны генерировать правдоподобные визуальные эффекты, их использование сопряжено с большим объемом вычислений, что делает подобные методы неприменимыми при визуализации масштабных сцен в реальном времени.

В моделях первой группы используется система уравнений Навье-Стокса. Одним из способов адаптации системы уравнений Навье-Стокса для визуализации воды на компьютере является дискретизация непрерывного потока жидкости путём введения двумерной или трёхмерной сетки и дальнейшее рассмотрение ее поведения как движения жидкости между ячейками сетки. Данный способ получил название метода сеток Эйлера.

Второй способ получил название метода Лагранжа и предлагает рассматривать жидкость как набор частиц, движущихся согласно физическим законам. Несмотря на то, что с помощью этого метода можно

получить реалистичные результаты, он не способен описать всю сложность внутреннего взаимодействия жидкости согласно уравнениям Навье-Стокса.

Параметрический и спектральный подходы вынесены в отдельную категорию как методы, пытающиеся объяснить поведение воды не только с физической точки зрения, однако применение их ограничено возможностью визуализации поверхности воды, когда расстояние до дна велико и погодные условия далеки от штормовых. Данный подход основывается на сложной математической модели океанских волн и также содержит большой объём вычислений, однако к нему применим алгоритм обратного быстрого преобразования Фурье, что позволяет в реальном времени создавать реалистичную поверхность океана.

Главная идея состоит в описании поверхности океана набором волн определённого спектра, полученного теоретическими исследованиями или путём измерений тех или иных параметров океана. Спектральный подход основан на возможности разложить карту высот поверхности океана на сумму гармонических волн [10]. Так, высота точки поверхности $y = h(\vec{p}, t)$, имеющей горизонтальную позицию $\vec{p} = (x, z)$, есть сумма синусоид с комплексными и зависящими от времени t амплитудами:

$$h(\vec{p}, t) = \sum_{\vec{k}} \tilde{h}(\vec{k}, t) e^{i\vec{k} \cdot \vec{p}}$$

где \vec{k} – двумерный волновой вектор;

$\tilde{h}(\vec{k}, t)$ – функция, определяющая структуру поверхности.

Эта функция выражается следующим образом:

$$\tilde{h}(\vec{k}, t) = h_0(\vec{k}) e^{i\omega t}$$

где ω – угловая частота волны с волновым вектором \vec{k} .

Основная трудоёмкость спектрального подхода заключается в подборе необходимой функции $h_0(\vec{k})$, чтобы результат визуализации обладал достаточной степенью реалистичности.

Pierson и Moskowitz [13] предложили концепт спектра волн океана, в котором волнения поверхности происходят за счёт ветра, дующего над всей поверхностью в определённом направлении в течение продолжительного времени. Проведя измерения, они выявили следующее соотношение, названное спектром Пирсона-Московица:

$$h_0(\vec{k}) = \frac{\alpha g^2}{16\pi^4 f^5} e^{-\frac{5}{4}(\frac{f_m}{f})^4}$$

где α – константа Филлипса ($\alpha = 0.0081$);

g – ускорение свободного падения;

f – частота волны с волновым вектором \vec{k} ;

f_m – пик частоты.

Пик частоты f_m зависит, в свою очередь, от ускорения свободного падения и от скорости ветра U_{10} , дующего на высоте 10 метров над поверхностью океана:

$$f_m = \frac{0.13g}{U_{10}}$$

К. Хассельман [7] внес улучшения в спектр Пирсона-Московица, $h_0(\vec{k}) = \frac{\alpha g^2}{16\pi^4 f^5} e^{-\frac{5}{4}(\frac{f_m}{f})^4} e^{\ln(\gamma) e^{-\frac{(f-f_m)^2}{2\sigma^2 f_m^2}}}$

где γ – числовая константа (обычно её берут равно 3.3);

$$\sigma = \begin{cases} 0.07, & f \leq f_m \\ 0.09, & f > f_m \end{cases} - \text{числовая константа.}$$

Несмотря на то, что результаты, полученные при помощи спектрального подхода, обладают высокой степенью визуальной реалистичности, с физической точки зрения они являются лишь приближением реальной картины физического мира. Высота точки поверхности воды $y = h(\vec{p}, t)$, имеющей горизонтальную позицию $\vec{p} = (x, z)$, может быть вычислена как сумма синусоид с комплексными и

зависящими от времени t амплитудами:

$$h(\vec{p}, t) = \sum_{\vec{k}} \tilde{h}(\vec{k}, t) e^{i\vec{k} \cdot \vec{p}} \quad (1)$$

где \vec{k} – двумерный волновой вектор;

$\tilde{h}(\vec{k}, t)$ – функция, определяющая структуру поверхности.

Поверхность воды можно дискретизировать следующим образом: взять горизонтальный кусок поверхности океана размером L_x метров на L_z метров, поместить в его середину начало системы координат, затем спроектировать на него двумерную дискретную сетку размером N на M точек с целочисленными координатами (начало системы координат помещается в центр сетки). Таким образом, теперь каждая точка сетки помимо собственных сеточных координат, будет иметь также и дискретные пространственные координаты – такая позиция точки обозначается как радиус-вектор \vec{p} :

$$\vec{p} = \left(\frac{nL_x}{N}, \frac{mL_z}{M} \right),$$

$$\text{где } -\frac{N}{2} \leq n < \frac{N}{2}, \\ -\frac{M}{2} \leq m < \frac{M}{2};$$

n и m – целочисленные координаты спроектированной двумерной сетки.

Введём также на данной сетке поле двумерных волновых векторов $\vec{k} = (k_x, k_z)$, имеющих дискретные пространственные координаты $k_x = \frac{2\pi n}{L_x}$ и $k_z = \frac{2\pi m}{L_z}$. Теперь можно записать (1) следующим образом:

$$h(\vec{p}, t) = \sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1} \sum_{m=-\frac{M}{2}}^{\frac{M}{2}-1} \tilde{h}(\vec{k}, t) e^{i\vec{k} \cdot \vec{p}} \quad (2)$$

где \vec{p} – дискретная пространственная позиция точки;

\vec{k} – двумерный волновой вектор с дискретными пространственными координатами k_x и k_z .

Функция $\tilde{h}(\vec{k}, t)$, определяющая структуру поверхности, Ю.Тессендорф [14] предложил представлять ее следующим образом:

$$\tilde{h}(\vec{k}, t) = \tilde{h}_0(\vec{k}) e^{i\omega(\vec{k})t} + \tilde{h}_0^*(-\vec{k}) e^{-i\omega(\vec{k})t}$$

где $\omega(\vec{k})$ – угловая частота волны с волновым вектором \vec{k} ;

\tilde{h}_0 – выбранная спектральная функция океанских волн;

\tilde{h}_0^* – комплексно сопряжённая к \tilde{h}_0 функция.

Значения $\omega(\vec{k})$ и \tilde{h}_0 рассчитываются следующим образом. Реалистичность анимированного поведения поверхности воды океана, создаваемого при помощи спектрального подхода, во многом зависит от набора угловых частот $\omega(\vec{k})$, соответствующих каждому \vec{k} из сгенерированного ранее поля двумерных волновых векторов. Из физической теории волн известно, что угловую частоту волны можно связать с длиной волнового вектора, соответствующего этой волне. В спектральном подходе, который моделирует поведение волн на поверхности океана в местах с большой глубиной, в этом соотношении фактор, учитывающий расстояние от поверхности до дна, можно без ущерба опустить, и прийти к следующей формуле:

$$\omega(\vec{k}) = g|\vec{k}|$$

где g – ускорение свободного падения.

Такое соотношение в теории волн называют законом дисперсии.

Далее определяется спектральная функция океанских волн \tilde{h}_0 как

$$\tilde{h}_0(\vec{k}) = \frac{1}{\sqrt{2}} (\xi_r + i\xi_i) \sqrt{P_h(\vec{k})}$$

где ξ_r и ξ_i – числа, полученные генератором случайных чисел с математическим ожиданием, равным 0, и стандартным отклонением, равным 1, подчинённые нормальному распределению;

$P_h(\vec{k})$ – спектр океанских волн Филлипса.

Спектр океанских волн Филлипса был получен на основе статистического анализа данных о волнах, вызываемых под влиянием ветра, и представляется в виде:

$$P_h(\vec{k}) = A \frac{e^{-(|\vec{k}|L)^{-2}}}{|\vec{k}|^4} \left| \frac{\vec{k} \cdot \vec{w}}{|\vec{k}| |\vec{w}|} \right|^2$$

где A – числовая константа, позволяющая регулировать высоту волн;

\vec{w} – двумерный вектор скорости ветра;

$L = \frac{v^2}{g}$ – максимальная высота волн, возникающих под влиянием непрерывного продолжительного ветра скорости V (g – ускорение свободного падения).

Стоит отметить, что в данной формуле множитель $\left| \frac{\vec{k} \cdot \vec{w}}{|\vec{k}| |\vec{w}|} \right|^2$ позволяет исключить влияние волн, движущихся перпендикулярно направлению ветра.

Модель поверхности океана не ограничивается вычислением высот дискретного набора точек, аппроксимирующих реальную поверхность океана. Для последующего вычисления освещённости поверхности океана необходимо находить нормали к точкам поверхности.

Положение точки поверхности океана в трёхмерном пространстве описывается её горизонтальной позицией $\vec{p} = (x, z)$ и высотой $h(\vec{p}, t)$, что позволяет записать её положение как

$$\vec{r}(\vec{p}, t) = \vec{p} + \vec{y} \cdot h(\vec{p}, t) \quad (3)$$

где \vec{y} – единичный вектор с координатам $(0, 1, 0)$ (то есть указывающий вверх).

Тогда в точке \vec{r} нормаль к поверхности вычисляется, используя градиент $\epsilon(\vec{p}, t) = \nabla h(\vec{p}, t)$:

$$\vec{n}(\vec{p}, t) = \frac{\vec{y} - \epsilon(\vec{p}, t)}{\sqrt{1 + \epsilon^2(\vec{p}, t)}}$$

Градиент вычисляется следующим образом:

$$\epsilon(\vec{p}, t) = \nabla h(\vec{p}, t) = \sum_{\vec{k}} i \vec{k} \tilde{h}(\vec{k}, t) e^{i \vec{k} \cdot \vec{p}}$$

Описанный выше метод генерации поверхности океана позволяет создавать волны лишь с закруглёнными гребнями, поскольку каждая из них является суммой большого числа гармонических волн, что создаёт эффект поведения океана в достаточно спокойную погоду. Однако в реальности дело обстоит несколько по-другому, и даже при достаточно хороших погодных условиях, а в особенности при ветре, океанские волны имеют заострённый гребень. Степень проявления данного эффекта в общем случае зависит от условий окружающей среды, длины волны и её высоты над поверхностью.

Поведение подобного рода описывается уравнениями Навье-Стокса, однако Д.Кример [3] разработал метод, позволяющий на основе использования спектрального подхода смоделировать эффект заострённых волн, не прибегая к прямому оперированию уравнениями. Данный метод в дополнение к генерации карты высот поверхности, приведённой в предыдущем пункте, вычисляет также карту смещений для каждой точки поверхности. Так, двумерный вектор смещения $\vec{D}(\vec{p}, t)$ для точки с горизонтальной позицией \vec{p} есть:

$$\vec{D}(\vec{p}, t) = \sum_{\vec{k}} -i \frac{\vec{k}}{|\vec{k}|} \tilde{h}(\vec{k}, t) e^{i \vec{k} \cdot \vec{p}}$$

Результат вычисления необходимо сложить с вектором исходных координат точки $\vec{p}_0 = (x_0, z_0)$, умножив на числовой коэффициент λ , регулирующий профиль получаемых волн: $\vec{p} = \vec{p}_0 + \lambda \cdot \vec{D}(\vec{p}, t)$.

Таким образом, теперь каждая точка поверхности с течением времени изменяет не только координату-высоту, но и координаты горизонтальной позиции. Предложенное решение не изменяет высоту волны непосредственно, а вместо этого подвергает деформации горизонтальные позиции точек поверхности так, чтобы эти изменения отразились на пространственной структуре карты высот поверхности. Деформации подобного рода позволяют заострять гребни волн и расширять их основания, а также являются деформациями нелинейного характера, что позволяет увеличить реалистичность финального изображения.

Используя алгоритм обратного быстрого преобразования Фурье, можно добиться эффективного вычисления карты высот поверхности воды океана при помощи формулы (1).

Для реализации описанного метода в качестве основного языка программирования был выбран C++. В качестве графического API для разработки был выбран OpenGL, в качестве математической библиотеки - платформо-независимая библиотека GLM (OpenGL Mathematics). Для создания оконного интерфейса, контекста OpenGL и управления вводом была выбрана кроссплатформенная библиотека GLFW.

Реализация алгоритма быстрого преобразования Фурье на GPU при помощи OpenGL Compute Shaders позволила значительно увеличить производительность приложения по следующим двум причинам: во-первых, данный алгоритм является адаптируемым к параллельному вычислению на GPU, и, во-вторых, результат вычислений хранится сразу в памяти видеокарты, а не переносится из оперативной памяти в память видеокарты после вычислений при помощи CPU. На рис. 1 приведено сравнение времени, требующегося на построение одного кадра сгенерированной поверхности воды (без вычисления нормалей и эффекта заострённых волн) при реализации алгоритма БПФ на CPU и GPU. Эксперимент проводился на компьютере с процессором Intel Core i7-2670QM 2.2GHz и видеокартой NVIDIA GeForce 630M (версия драйвера видеокарты 364.72).

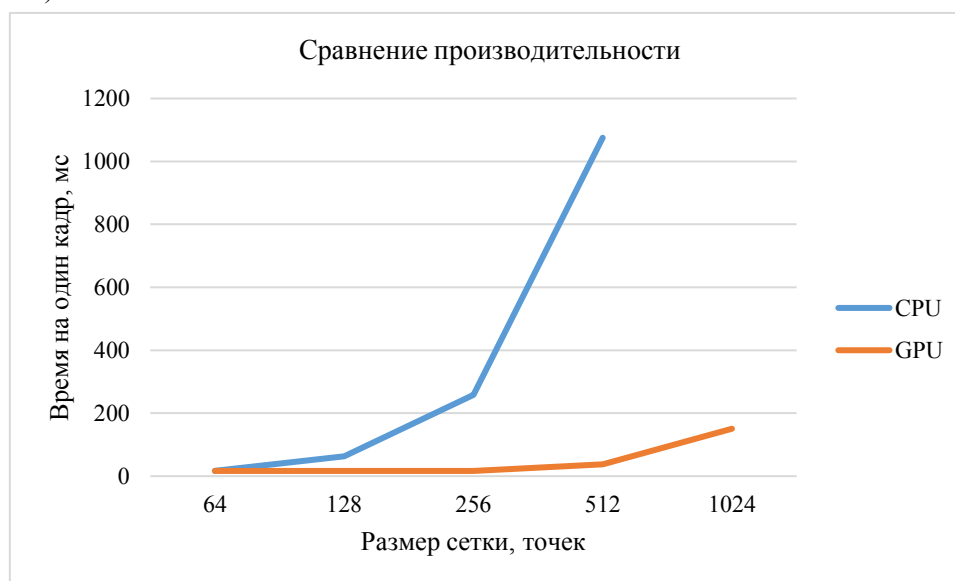


Рисунок 1 – Сравнение производительности БПФ на CPU и GPU без учета вычисления нормалей и эффекта заостренных волн

Отметим, что при малых размерах сетки (общее количество точек $\leq 64 * 64$) алгоритм БПФ эффективно работает как на CPU, так и на GPU, однако с дальнейшим увеличением точек в сетке производительность алгоритма на CPU начинает значительно уступать, причём уже для сетки с общим числом точек $\geq 128 * 128$ визуализация происходит с заметным для глаза сниженным количеством кадров в секунду.

Реализация на GPU начинает работать медленнее для сетки размером $\geq 1024 * 1024$ точек. В табл. 1 приведены полные результаты эксперимента.

Время, требуемое на рендеринг одного кадра сгенерированной поверхности воды при реализации алгоритма БПФ на CPU и GPU.

Размер сетки	16	32	64	128	256	512	1024
Время CPU, мс	16,6	16,6	16,9	63,4	258,3	1075,3	4545,5
Время GPU, мс	16,6	16,6	16,6	16,6	16,6	37,5	150,7

На рис. 2 представлен пример работы программы в режиме генерации сетки размером 64 * 64 точек..

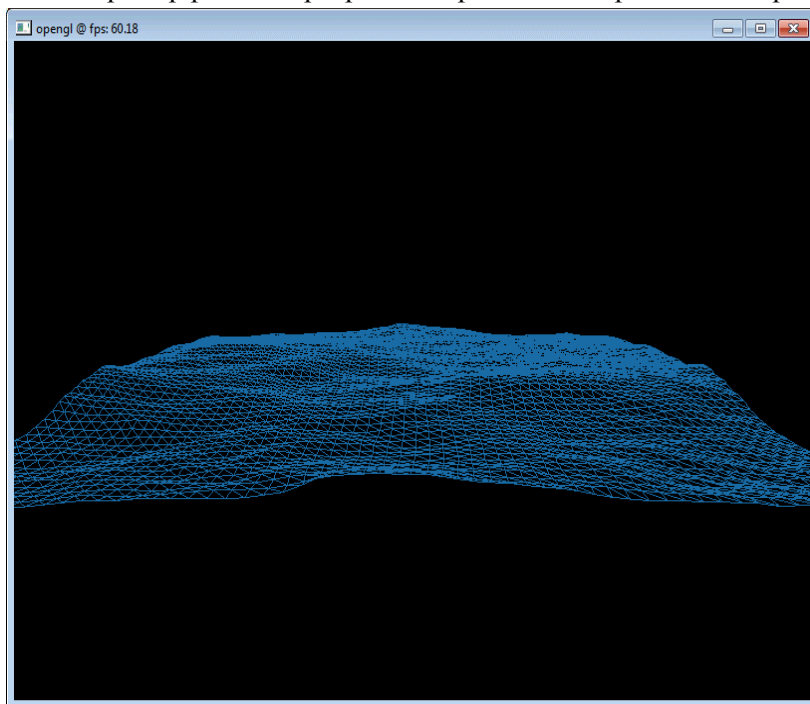


Рисунок 2 – Работа программы в режиме генерации сетки размером 64 * 64 точек.

На рис.3 поверхность имеет тот же размер, но отображается в виде набора треугольников с наложенным освещением

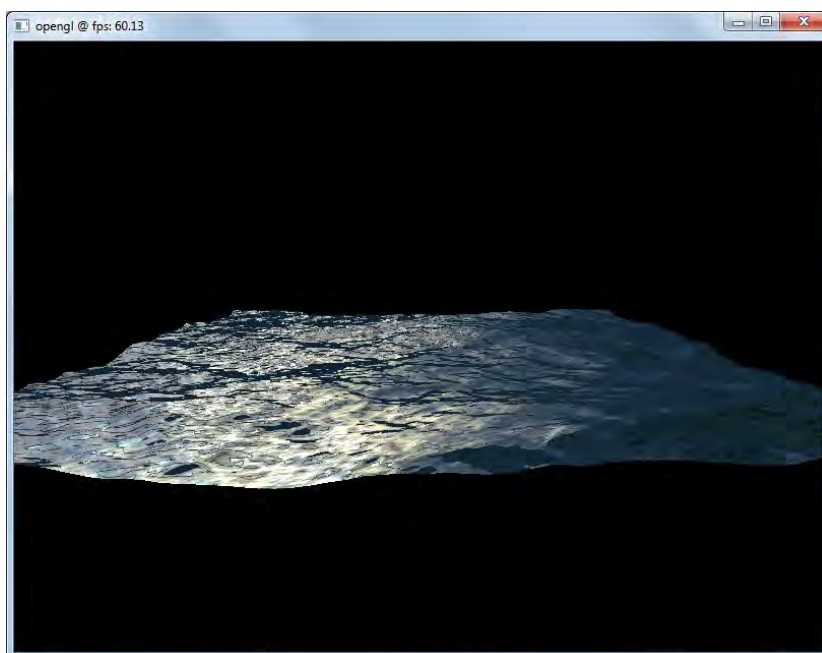


Рисунок 3 – Работа программы с отображением поверхности в виде набора треугольников с наложенным освещением.

Список использованной литературы:

1. Bailey, M. (2016). OpenGL Compute Shaders. [Online]. Available: <http://web.engr.oregonstate.edu/~mjb/cs557/Handouts/compute.shader.1pp.pdf>. [Accessed: 21.04.2016].
2. Cormen, T. (2009). Introduction to algorithms. Cambridge, Mass.: MIT Press.
3. Creamer, D., Henyey, F., Schult, R. and Wright, J. (1989). Improved linear representation of ocean surface waves. J. Fluid Mech., 205(-1), p.135.
4. Fournier, A. and Reeves, W. (1986). A simple model of ocean waves. ACM SIGGRAPH Computer Graphics, 20(4), pp.75-84.
5. Glfw.org. (2016). GLFW. [Online]. Available: <http://www.glfw.org>. [Accessed: 15.02.2016]
6. Glm.g-truc.net. (2016). OpenGL Mathematics. [Online]. Available: <http://glm.g-truc.net/0.9.7/index.html>. [Accessed: 08.03.2016].
7. Hasselman K., Barnett T., Bouws E., Carlson D. E., Hasselman P. (1973). Measurements of wind-wave growth and swell decay during the joint north sea wave project (jonswap). Deutsche Hydrographische Zeitschrift, Deutsches Hydrographisches Institut, Hamburg, Germany, A8, 12.
8. Jr, N. (2015). OpenGL SuperBible. [S.l.]: Addison-Wesley Professional.
9. Khronos.org. (2016). The open standard for parallel programming of heterogeneous systems. [Online]. Available: <https://www.khronos.org/opencl>. [Accessed: 19.04.2016].
10. Li, Z. and Hou, X. (2014). Ocean Wave Real-Time Simulation Based-on Ocean Wave Spectrum and FFT. AMR, 926-930, pp.3531-3536.
11. Opengl.org. (2016). Compute Shader. [Online]. Available: https://www.opengl.org/wiki/Compute_Shader. [Accessed: 19.04.2016].
12. Opengl.org. (2016). OpenGL Shading Language. [Online]. Available: <https://www.opengl.org/documentation/glsl/>. [Accessed: 25.03.2016].
13. Pierson, W. and Moskowitz, L. (1964). A proposed spectral form for fully developed wind seas based on the similarity theory of S. A. Kitaigorodskii. J. Geophys. Res., 69(24), pp.5181-5190.
14. Tessendorf J. (2001). Simulating ocean waters. ACM SIGGRAPH Computer Graphics.
15. Tian, L. (2014). Ocean wave simulation by the mix of FFT and Perlin Noise. WSCG, 26(2), pp. 41-45.
16. Wolff, D. (2013). OpenGL 4 Shading Language Cookbook. Birmingham: Packt Publishing.

© Куров А.В., Завалин А.А. 2017

УДК 004.8**Лавренков Юрий Николаевич**канд. техн. наук, доцент КФ МГТУ им. Н.Э. Баумана, г. Калуга, РФ,
e-mail: georglawr@yandex.ru**Цыганков Иван Сергеевич**студент 4 курса КФ МГТУ им. Н.Э. Баумана, г. Калуга, РФ,
e-mail: georglawr@yandex.ru**СИНХРОНИЗАЦИЯ ПОПУЛЯЦИИ НЕЙРОНОВ С ПРИМЕНЕНИЕМ НЕЙРОСЕТЕВОГО
ГЕНЕРАТОРА СТОХАСТИЧЕСКИХ ВОЗМУЩЕНИЙ****Аннотация**

Оценивается возможность применения нейросетевого генератора стохастических возмущений для синхронизации популяции нейронов, реализованных в виде осциллирующих генераторов. Корреляционные синапсы, используемые для связи элементов популяции, осуществляют функцию фильтрации случайных возмущений, изменяющих порог активации нейрона. В результате воздействия на нейроны популяции