

Моделирование волн на поверхности жидкости

© А.В. Куров, К.А. Якиль

МГТУ им. Н.Э. Баумана, Москва, 105005, Россия

Рассмотрены методы моделирования волн на поверхности жидкости, включая круговую волну и наложение волн. Предложена программная реализация визуализации волн, основанная на использовании алгоритма построчного сканирования и простой модели освещения Фонга. Рассмотрены способы увеличения скорости работы алгоритма.

Ключевые слова: *поверхность воды, круговые волны, волны, наложение волн, построчный z-буфер, модель освещения.*

Задача визуализации поверхности жидкости (в частности воды) в реальном времени является актуальной, так как широко используется в построении изображений в компьютерных играх и при моделировании спецэффектов. В настоящее время существует большое количество систем моделирования жидкости, наиболее популярными из которых являются RealFlow, Blender, Glu3d, AfterBurn, FumeFX. Данные системы основываются на гидродинамике сглаженных частиц (англ. smoothed particle hydrodynamics — SPH) [2], стабильном методе решеточных уравнений Больцмана [3] или на методе несжимаемых уравнениях Эйлера для массы и сохранения импульса.

Перечисленные подходы позволяют реалистично визуализировать поверхность воды, они учитывают физические характеристики отдельных частей и предусматривают вычисление силы их взаимодействия либо используют физические зависимости между характеристиками целого объекта.

Программная реализация таких алгоритмов требует значительного времени выполнения, большого объема памяти и существенным образом загружает центральный процессор (ЦП). Поэтому для поддержания реального времени задействована работа с аппаратной частью.

В статье рассматривается разработка программы, которая требует меньшего объема памяти, обеспечивает меньшую загрузку ЦП, реализует отображение в реальном времени за счет выбора соответствующего метода моделирования и алгоритма визуализации, при этом отсутствует непосредственное взаимодействие с аппаратной частью.

Алгоритм задания сетки. Изначально для представления поверхности жидкости используется каркасная (сеточная) модель. Для быстрого расчета новых координат сетки применяется исходный массив

точек (вершин сетки), который далее разбивается на список треугольников. Стороны плоскости параллельны координатным осям.

Для вычисления значений координат точек поверхности жидкости на каждом шаге в любой момент времени необходимо знать волновое уравнение.

Задание уравнения бегущей волны. Бегущие волны моделируются свободными гармоническими колебаниями [2, 3]. Для перехода в нормализованное пространство, в котором значение функции лежит в пределах от 0 до 1, выполняется преобразование:

$$f(x) = \frac{\sin x + 1}{2}. \quad (1)$$

С целью получения реалистичного изображения волн необходимо принять во внимание тот факт, что волны могут иметь большую крутизну и остроту пиков. С учетом амплитуды колебаний получим

$$f(x) = A \left(\frac{\sin x + 1}{2} \right)^k.$$

Необходимо также учитывать направление волны. Ее моделирование происходит в двумерном поле высот, поэтому требуется определять движение волн в обоих направлениях. Вектор направления должен быть параллельным плоскости невозмущенной поверхности жидкости, т. е. иметь координату z , равную нулю.

$$S = Dir(x, y) \cdot Pos(x, y), \quad (2)$$

где $Dir(x, y)$ — вектор направления волны, $Pos(x, y)$ — вектор координат точки.

Учет частоты происходит в формуле в соответствии с известным соотношением определения частоты f :

$$f = \frac{2\pi}{\lambda},$$

где λ — длина волны.

Умножая (2) на частоту, получим:

$$S = Dir(x, y) \cdot Pos(x, y) f.$$

Для учета скорости волны необходимо определить фазовую постоянную в соответствии с выражением

$$\phi = \nu f = \frac{2\nu\pi}{\lambda}.$$

Окончательное выражение для вычисления координаты точки z на поверхности жидкости в зависимости от ее координат x, y и времени имеет следующий вид:

$$f(x, y, t) = A \left(\frac{\sin S + 1}{2} \right)^k,$$

где $S = \text{Dir}(x, y) \cdot \text{Pos}(x, y) f + t\phi$.

Наложение волн. С целью получения более реалистичного изображения волн возникает необходимость рассмотрения нескольких волн и их наложения, как на самом деле происходит на поверхности жидкости. Учитывая принцип суперпозиции и уравнение отдельной волны, итоговое значение функции может быть вычислено как сумма значений функций, описывающих отдельные волны. Итоговое уравнение:

$$H(x, y, t) = \sum_{j=1}^l (A_j \sin(D_j(x, y) f_j + t\phi_j)),$$

где A_j — амплитуда j -й бегущей волны; D_j — вектор направления распространения j -й бегущей волны; f_j — частота j -й бегущей волны; t — время; ϕ_j — фазовая постоянная j -й бегущей волны.

В такой модели есть возможность наблюдать интерференцию волн.

Круговые волны. Наряду с наложением волн, моделирование круговых волн также улучшает реалистичность создаваемого изображения поверхности жидкости. Данный вид волн позволяет наблюдать поведение поверхности жидкости в случае эффекта «брошенного камня».

Учесть этот эффект можно с помощью переопределения вектора направления в каждой точке поверхности и изменения координаты точки смещением начала координат в точку центра круговой волны:

$$D_i(x, y) = \left(\frac{(x, y) - C_i}{|(x, y) - C_i|} \right),$$

$$X_n = X - X_c,$$

$$Y_n = Y - Y_c,$$

где C_i — точка начала распространения i -й круговой волны; X, Y — координаты рассматриваемой точки; X_n, Y_n — преобразованные координаты точки; X_c, Y_c — координаты точки начала распространения волны.

Измененные значения подставляются в формулу бегущей волны.

Вычисление нормалей. При визуализации поверхности жидкости производится переход от каркасной модели к поверхностной, причем для изображения получаемой поверхности необходимо вы-

числять интенсивность освещения, что, в свою очередь, требует знания вектора нормали в точках поверхности. Используя уравнения поверхностей, определим нормали в любой ее точке. Уравнение нормали к поверхности в точке (x, y, z) :

$$N(x, y, z) = NB(x, y, z) \otimes NT(x, y, z),$$

где $NB(x, y, z)$ и $NT(x, y, z)$ — частные производные по x и y ;

$$NB(x, y, z) = \frac{\partial f(x, y, t)}{\partial x},$$

$$NT(x, y, z) = \frac{\partial f(x, y, t)}{\partial y},$$

где $f(x, y, t)$ — итоговое уравнение поверхности;

$$\frac{\partial f(x, y, t)}{\partial x} = 0,5kDir_x fA \left(\frac{\sin S + 1}{2} \right)^{k-1} \cos S,$$

$$\frac{\partial f(x, y, t)}{\partial y} = 0,5kDir_y fA \left(\frac{\sin S + 1}{2} \right)^{k-1} \cos S.$$

Задавая векторы нормалей проекциями на координатные оси, получим

$$NB(x, y, z) = \left(\frac{\partial x}{\partial x}, \frac{\partial y}{\partial x}, \frac{\partial f(x, y, t)}{\partial x} \right) = \left(1, 0, \frac{\partial f(x, y, t)}{\partial x} \right),$$

$$NT(x, y, z) = \left(\frac{\partial x}{\partial y}, \frac{\partial y}{\partial y}, \frac{\partial f(x, y, t)}{\partial y} \right) = \left(0, 1, \frac{\partial f(x, y, t)}{\partial y} \right).$$

Поскольку производится поиск общего решения, то необходимо найти частные производные от сумм волновых уравнений по x и y . Итоговая формула для вычисления нормали

$$N(x, y) = \left(-\sum_{j=0}^l \frac{\partial f(x, y, t)}{\partial x}, -\sum_{j=0}^l \frac{\partial f(x, y, t)}{\partial y}, 1 \right).$$

Выбор алгоритма визуализации. На реалистичность изображения влияет модель освещения. Поскольку необходимо поддерживать визуализацию изображения в реальном времени, выбрана простая модель освещения, в которой учитывается диффузное отражение, рассеянный свет и используется упрощенная модель Буи-Туонга Фонга для зеркального отражения [1]:

$$I = I_p k_p + \sum_{j=1}^m \frac{I_{wj} (k_d \cos \theta_i + k_s \cos^n \alpha_j)}{d_j + L},$$

где I — интенсивность освещения в точке; I_p — интенсивность рассеянного света; k_p — коэффициент диффузного отражения рассеянного света; I_{ij} — интенсивность j -го источника света; k_d — коэффициент диффузного отражения; θ_j — угол, образованный вектором падения и вектором нормали; k_z — коэффициент зеркального отражения; α_j — угол, образованный вектором отражения и вектором нормали; n — степень, аппроксимирующая пространственное распределение зеркально отраженного света; d_j — расстояние от точки поверхности до j -го источника света (обычно для всех источников это расстояние отсчитывается от точки поверхности до центра проекции); L — постоянная, подбираемая из условия получения более реалистичного изображения; m — количество источников освещения.

Визуализация поверхности жидкости. Для визуализации поверхности жидкости за основу был взят алгоритм строчного сканирования, использующий Z -буфер [1]. Поверхность жидкости аппроксимируется треугольниками. Преимущества данного представления: простое разбиение массива точек на треугольники, обеспечение хорошей реалистичности отображения поверхности воды вследствие возможности детального моделирования рельефной поверхности, удобство в использовании треугольников в алгоритме строчного сканирования, использующего Z -буфер, из-за сведения оперирования многоугольниками к оперированию треугольниками.

При реализации в алгоритм были внесены некоторые изменения, направленные на его упрощение и повышение быстродействия. Так как задание поверхности жидкости осуществляется набором треугольников, то заранее известно, что никаких других многоугольников во входных данных не поступает. Данное допущение позволяет:

1) не запоминать характеристики треугольника в дополнительном списке характеристик, а добавить необходимые данные в структуру ребра. Таким данным является конечная точка ребра. Это преобразование ведет к ускорению работы алгоритма;

2) не хранить список активных многоугольников, так как работа ведется с треугольниками, которые представлены как активные ребра с дополнительной информацией — конечной точкой ребра. Это позволяет экономить память и время, затрачиваемое на обработку списка. Активные ребра изначально заносятся в пару активных ребер в правильном порядке;

3) не выяснять принадлежность многоугольника, ребро которого перестало быть активным, к списку активных многоугольников.

Структура активного ребра дополняется точным значением конечной точки ребра.

Такой подход обладает следующими преимуществами. Изменение списка активных ребер происходит быстрее, так как имеется информация о ребре, которое необходимо добавить.

Пусть есть пара ребер E_1 и E_2 и необходимо удалить ребро E_1 . Тогда добавляемое ребро есть P_1P_2 , где P_1 — конечная точка ребра E_1 , P_2 — конечная точка ребра E_2 .

Для реализации возможности вычисления нормалей в точке с последующим определением ее цвета необходимо знать координату точки в мировой системе координат (расчет интенсивности происходит именно в ней).

Для этого вводится дополнительный список активных ребер, хранящий информацию о ребре в мировой системе координат.

Структура такого ребра:

- 1) текущая точка;
- 2) вектор приращения значения каждой из координат точки;
- 3) текущее значение нормали;
- 4) вектор приращения значения каждой из координат нормали.

Данный подход требует больше памяти для хранения такой структуры, но увеличивает скорость работы, так как не приходится вычислять значение нормали в каждой точке ребра, что является трудоемкой операцией. Кроме того, обработка такой структуры происходит в том же порядке и в то же время, в которое происходит обработка активных ребер (значения которых соответствуют экраным координатам), поэтому нет необходимости отдельно обрабатывать такие структуры и, соответственно, тратить процессорное время.

При расположении части сетки за пределами верхней части экрана возникает «ребристый край» сетки. Это происходит из-за того, что те треугольники, чьи минимальные значения координат у вершин имеют отрицательное значение, не обрабатываются. В программе массив Y -групп расширяется на отрицательные значения, что позволяет корректно отображать поверхность.

В программе реализован учет прозрачности.

Использование общеизвестного алгоритма построчного сканирования, использующего Z -буфер, при учете эффекта прозрачности ведет к возникновению ряда некорректных ситуаций:

- 1) в вершинах треугольников сетки резко увеличивается интенсивность;
- 2) при экранировании одной волной других волн наблюдается резкое увеличение интенсивности.

Первая ситуация связана с многократной обработкой вершины треугольника, так как она является общей сразу для нескольких треугольников. Поэтому в программе обрабатывается только интервал $[a;b]$, где a — точка левого ребра, b — точка правого ребра.

Правое ребро будет обработано соседним справа треугольником. Кроме того, вершина треугольника, имеющая минимальную координату по y , будет обработана не этим треугольником, а располагающимся на экране выше.

При многократном наложении прозрачных треугольников увеличивается коэффициент в буфере коэффициентов прозрачности. Поэтому в программе выполняется следующая проверка: если сумма текущего значения коэффициента прозрачности превышает заданный предел, то значение буфера коэффициентов и интенсивности не изменяется.

Для обеспечения большего быстродействия в программе не используется установка значения цвета пиксела методом `Bitmap.SetPixel(int x,int y,Color SetColor)` из-за его медленной работы. Вместо него прodelывают работу с классом `BitmapData` до начала работы алгоритма. Данный класс позволяет преобразовать значения альфа-канала и RGB-компонент в одномерный массив байтов. Использование массива не увеличивает существенно образом время работы алгоритма, так как можно получить быстрый доступ к любому элементу массива, вычисляя его адрес по индексу. После этапа формирования массив передается в программу, реализующую алгоритм закраски. Данное преобразование значительно ускоряет работу алгоритма.

На рис. 1–7 приведены результаты работы программы.

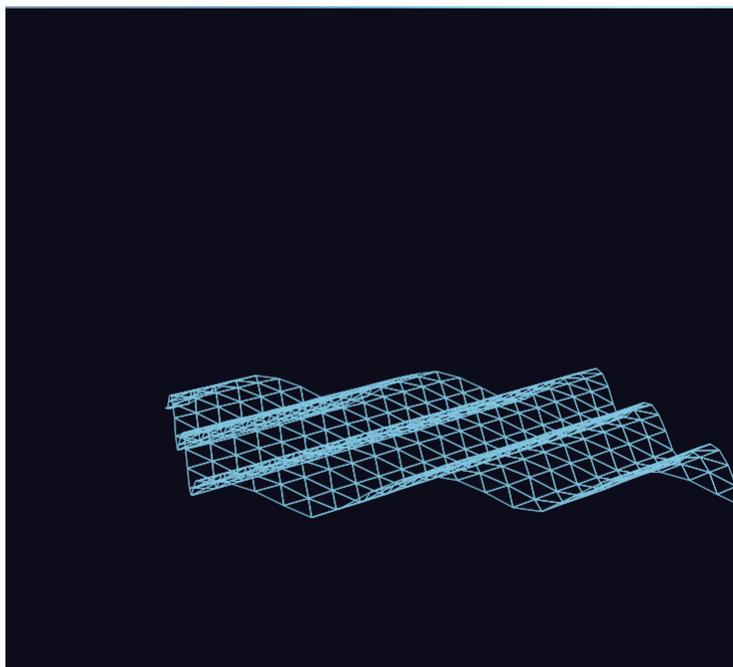


Рис. 1. Каркасная модель бегущей волны

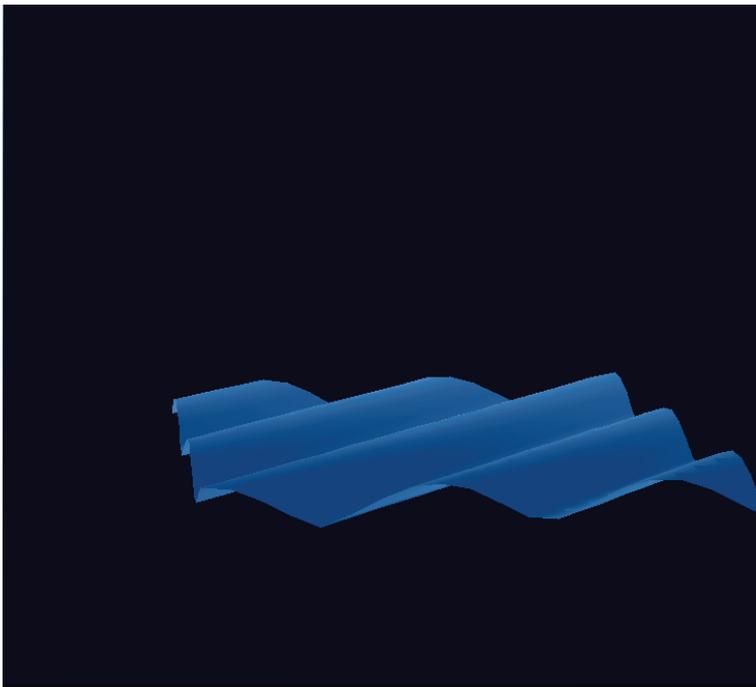


Рис. 2. Бегущая волна с закраской

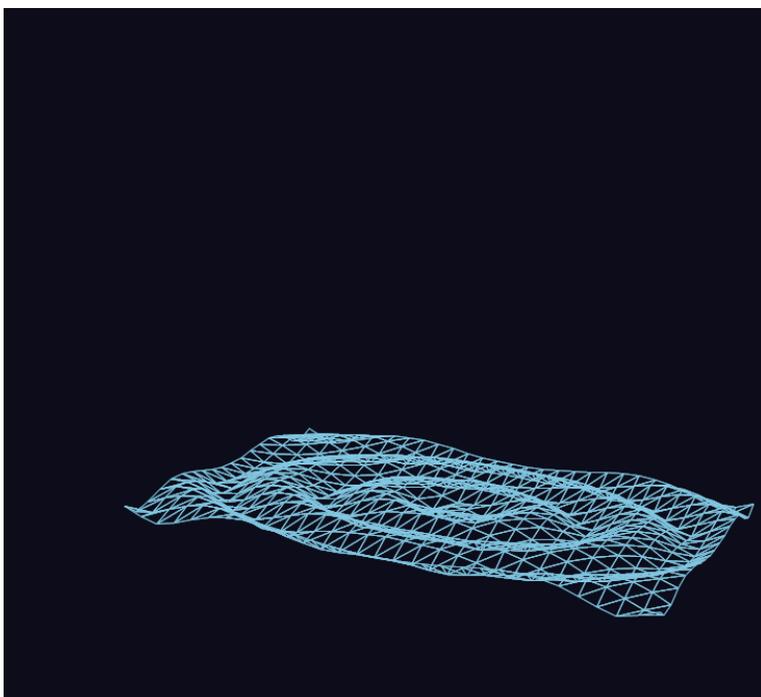


Рис. 3. Каркасная модель круговой волны

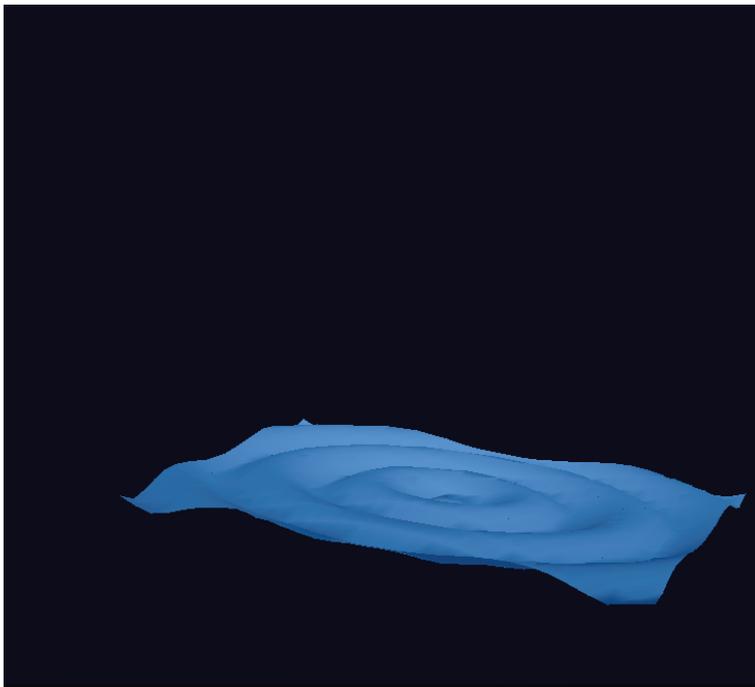


Рис. 4. Круговая модель с закраской

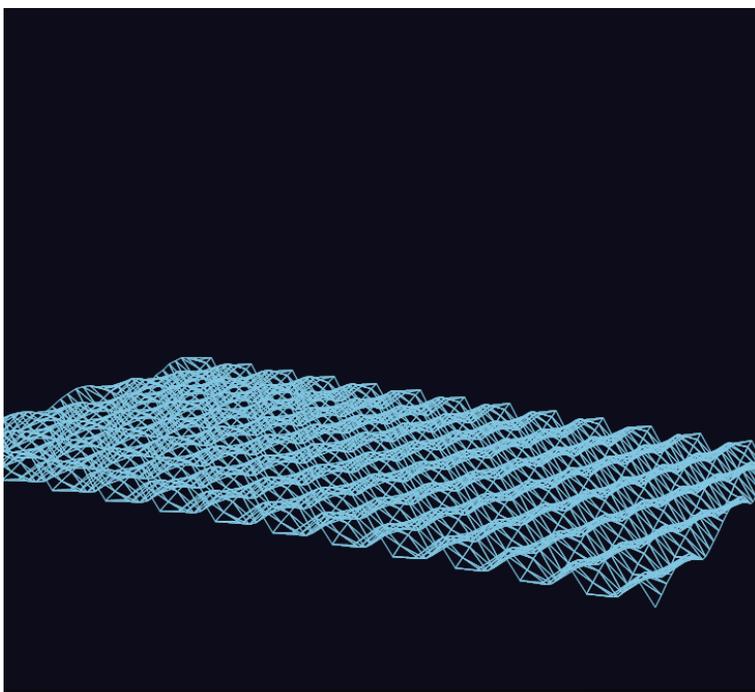


Рис. 5. Каркасная модель наложения бегущих волн

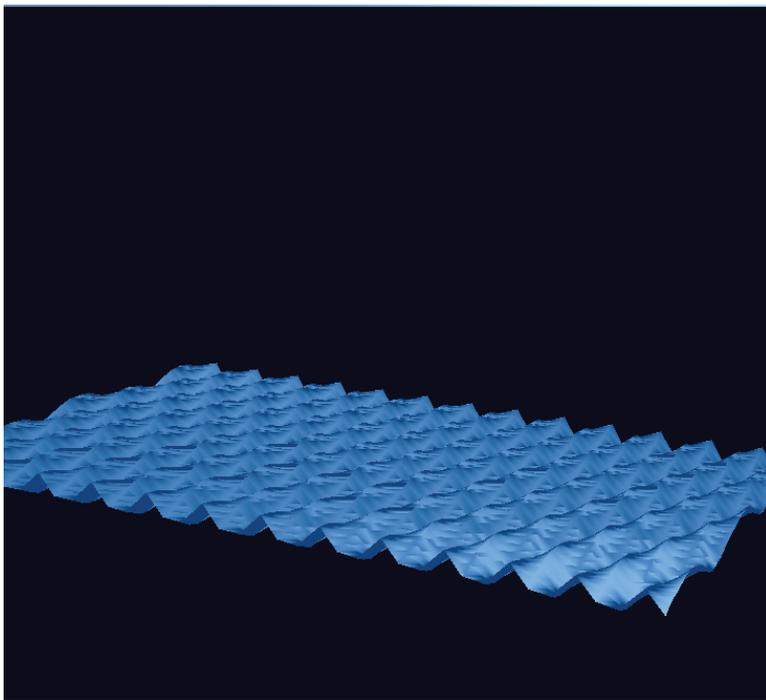


Рис. 6. Наложение бегущих волн с закраской

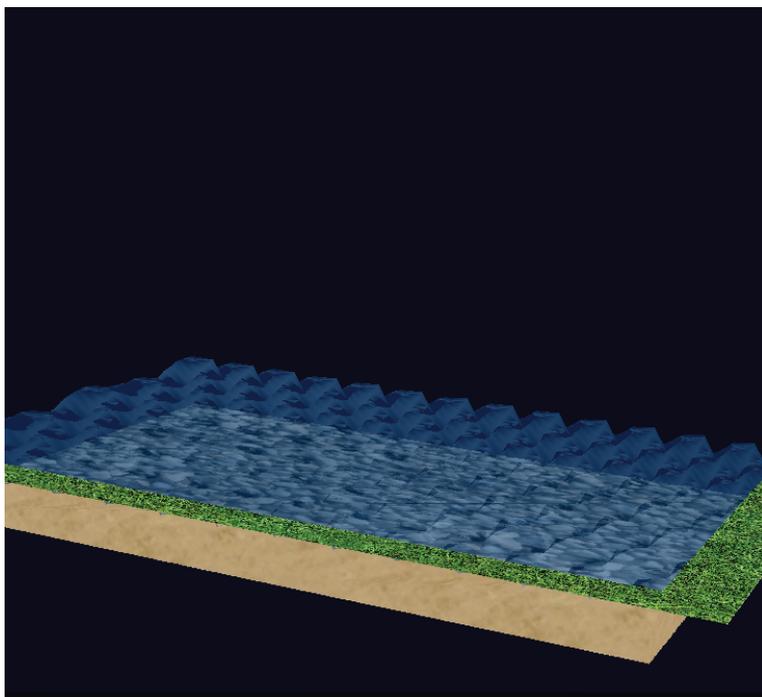


Рис. 7. Наложение бегущих волн с закраской с учетом прозрачности

ЛИТЕРАТУРА

- [1] Роджерс Д. *Алгоритмические основы машинной графики*. Москва, Мир, 1989, 512 с.
- [2] NVIDIA Developer Zone [Электрон. ресурс]. — Режим доступа <https://developer.nvidia.com/content/gpu-gems-chapter-1-effective-water-simulation-physical-models>
- [3] Intel Software [Электрон. ресурс]. — Режим доступа <http://software.intel.com/ru-ru/articles/real-time-deep-ocean-simulation-on-multi-threaded-architectures>
- [4] Википедия. Свободная энциклопедия [Электрон. ресурс]. — Режим доступа <http://ru.wikipedia.org/wiki>

Статья поступила в редакцию 10.06.2013

Ссылку на эту статью просим оформлять следующим образом:

Куров А.В., Якиль К.А. Моделирование волн на поверхности жидкости. *Инженерный журнал: наука и инновации*, 2013, вып. 6. URL: <http://engjournal.ru/catalog/it/hidden/773.html>

Куров Андрей Владимирович родился в 1955 г., окончил МВТУ им. Н.Э. Баумана в 1978 г. Канд. техн. наук, доцент кафедры «Программное обеспечение ЭВМ и информационные технологии» МГТУ им. Н.Э. Баумана. Автор более 50 научных трудов в области надежности вычислительных систем и программного обеспечения ЭВМ. e-mail: avkur7@mail.ru

Якиль Ксения Андреевна родилась в 1993 г. Студентка 3-го курса кафедры «Программное обеспечение ЭВМ и информационные технологии» МГТУ им. Н.Э. Баумана. e-mail: tricia@ngs.ru