

# Тестирование вычислительных возможностей метода SPH для моделирования физических эффектов в компьютерной графике

Светлана Александровна Кукаева, Вадим Евгеньевич Турлапов  
Нижегородский государственный университет им.Н.И. Лобачевского, Н.Новгород, Россия  
kukaeva\_svetlana@mail.ru, vadim.turlapov@cs.vmk.unn.ru

## Abstract

Modeling of liquid and heat exchange using method SPH is considered. Structure for acceleration was realized (uniform grid). Few effects: Falling of liquid column and solid body to liquid, heat exchange in body and melting, are numerically solved. Speed of work on CPU dependence from number of particles was investigated. Results of numeric computations are presented. Results allow to modeling about 2000-3000 particles in real-time on CPU.

**Key Words:** SPH, modeling of liquid body, heat exchange, ice melting

## 1. ВВЕДЕНИЕ

Существует ряд задач, который имеет большое прикладное значение для компьютерной графики: моделирование жидкости, моделирование взаимодействия двух тел, исследование процесса теплопередачи. Известно большое количество методов, используемых для этой цели. В данной работе за основу взят метод Гидродинамики Сглаженных Частиц (SPH – Smoothed Particle Hydrodynamics), как единственный метод решения таких задач. Его применяют для исследования больших деформаций эластичных тел [1], процессов теплопередачи и отвердевания [2], моделирования жидкостей и газов [3], [4], [5], [6], и др. Все это применяется в игровой индустрии, спецэффектах кино и т.д. Цель настоящей работы – реализовать для многоядерных CPU и протестировать вычислительные возможности метода SPH для моделирования перечисленных выше физических эффектов в компьютерной графике.

## 2. ТЕСТИРОВАНИЕ МЕТОДА SPH

Метод SPH (разработан Люси в 1977, и независимо Джинголом и Монаганом в 1977) - лагранжевый метод, в котором для моделирования используются частицы, для каждой из которых в любой момент времени известны масса, положение, скорость, а также плотность и давление.

Итак, пусть у нас есть частица  $x_i$  с массой  $m_i$  и некоторым свойством  $A_i$ . Тогда, если известны значения этого свойства для частиц, находящихся в некоторой окрестности  $x_i$  (в пределах так называемых двух сглаживающих длин), то  $A_i$  будет вычисляться как

$$(1) \quad A(x) = \sum_j m_j \frac{A_j}{\rho_j} W(x - x_j, h)$$

Где  $\rho_j$  - плотность частицы,  $W$  – сглаживающая функция,  $h$  – длина сглаживания. В формуле (1) и далее предполагается, что суммирование ведется по всем частицам, находящимся в пределах двух сглаженных длин.

Сглаживающая функция такова, что

\* на процессоре Intel(R) Celeron(R) CPU 2.6GHz 768 Mb RAM

\*\* на процессоре Intel(R) Core(TM)2 E6600 2.4GHZ, 3.25 Gb RAM

$$\int_{x-x_i} W(x - x_i, h) dx = 1$$

$$W(x - x_i, h) = 0, |x - x_i| > 2h$$

В качестве тестовых в данной работе приняты задачи падения столба жидкости и тела (более подробно алгоритм решения описан в [4], [5]), теплопередача, таяние льда [2]. В рамках решения данных задач решаются уравнения Навье-Стокса (2) и теплопередачи (3).

$$(2) \quad \frac{dv}{dt} = -\frac{\nabla p}{\rho} + \mu \frac{\nabla^2 v}{\rho} + \frac{F}{\rho}$$

$$(3) \quad c_p \frac{dT}{dt} = \frac{1}{\rho} \nabla(k \nabla T)$$

В классической реализации данные алгоритмы имеют оценку сложности  $O(n^2)$ .

Таблица 1. Результаты тестирования классического метода SPH

Количество частиц	FPS*		
	Падение столба жидкости	Теплопередача	Таяние льда
1000	20	16	14
1500	14	10	9
2000	7	6	4
2500	6	4	3
3000	4	3	2

Таблица 2. Результаты тестирования метода SPH с использованием OpenMP

Количество частиц	FPS**		
	Падение столба жидкости	Теплопередача	Таяние льда
1000	47	46	47
1500	31	24	33
2000	25	21	25
2500	20	15	14
3000	15	13	11

Для ускорения вычислений использовался алгоритм, основанный на равномерной сетке. Пусть моделируемая сцена разбита на  $k$  ячеек сетки. В этом случае для каждой частицы  $i$  необходимо просмотреть все частицы, находящиеся в одной и в соседних ячейках с ней

Таблица 3. Зависимость скорости работы программы от среднего числа частиц в одной ячейке сетки\*.

1000	500	100	50	20
20	20,5	21	22	24

Пусть в среднем в каждой ячейке находится  $s$  частиц. Тогда для расчета всех сил, действующих на каждую

частицу, необходимо затратить время порядка  $O(nc)$ . Чем больше величина  $c$ , тем хуже эта оценка. Следовательно, лучших (по вычислительным затратам и по точности) результатов можно добиться в том случае, когда сторона одной ячейки сетки принимается равной  $2h$ . Для данной реализации алгоритма так же были протестированы результаты.

Таблица 4. Результаты тестирования метода SPH с использованием равномерной сетки

Количество частиц	FPS*		
	Падение столба жидкости	Теплопередача	Таяние льда
1000	24	19	21
1500	14	11	12
2000	9	8	7
2500	6	6	6
3000	5	5	4

Таблица 5. Результаты тестирования метода SPH с использованием равномерной сетки и OpenMP

Количество частиц	FPS**		
	Падение столба жидкости	Теплопередача	Таяние льда
1000	47	47	45
1500	33	31	28
2000	28	25	23
2500	22	20	17
3000	17	18	16

В следующей таблице приведены обобщенные результаты тестирования метода SPH.

Таблица 6. Средняя оценка вычислительных возможностей метода SPH

Количество частиц	FPS**			
	Без использования OpenMP		С использованием OpenMP	
	Без сетки	С сеткой	Без сетки	С сеткой
1000	26	29	46,7	46,3
1500	16	19	29,3	30,7
2000	12,3	16	23,7	25,3
2500	9	11,9	16,3	19,7
3000	7,3	8,7	13	17

## 2.1. Результаты моделирования

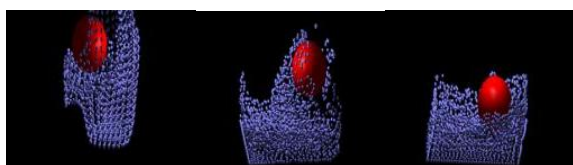


Рисунок 1. Падение тела, столба жидкости, 1800 частиц (см. табл.5)

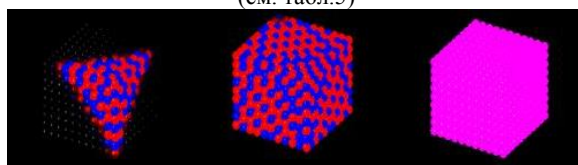


Рисунок 2. Теплопередача в теле, 1089 частиц (см. табл.5). Температура тела показана цветом, синим цветом обозначены участки с более низкой температурой, красным – с более высокой. Нагревание ведется от точечного источника тепла

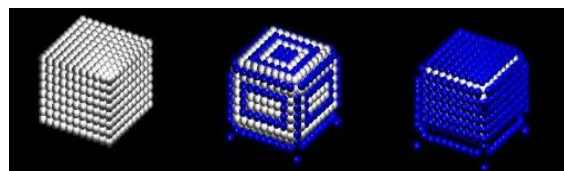


Рисунок 3. Таяние льда, 1089 частиц (см. табл.5). Белым цветом показан лед, синим – образовавшаяся в ходе таяния вода. Нагревание происходит в результате контакта с воздухом.

## 3. ЗАКЛЮЧЕНИЕ

Реализован алгоритм SPH, позволяющий также использовать ускоряющую структуру (равномерную сетку), для решения задач падения столба жидкости, исследования процессов теплопередачи, таяния льда. Протестированы вычислительные возможности реализованного алгоритма на CPU в зависимости от исследуемой проблемы. Для визуализации использовались простейшие средства OpenGL. Представлены результаты вычислительных экспериментов. Показано, что в реализации на CPU с использованием многоядерности, метод SPH может обеспечить реальное время моделирования во всех рассмотренных задачах примерно для 2000-3000 частиц. Результаты немного превосходят аналогичные исследования [7]. Результаты работы [3] показывают, что реализация SPH метода на GPU способна увеличить производительность в несколько раз.

## 4. СПИСОК ЛИТЕРАТУРЫ

- [1] Yoichi Kawashima and Yuzuru Sakai. Large Deformation Analysis of Hyperelastic Materials Using SPH Method, e-Journal of Soft Materials, Vol. 3(2007), 21–28
- [2] Joseph J. Monaghan, Herbert E. Huppert, M. Grae Worster. Solidification using smoothed particle hydrodynamics, Journal of Computational Physics 206 (2005), 684–705
- [3] Yanci Zhang, Barbara Solenthaler, Renato Pajarola. Adaptive Sampling and Rendering of Fluids on the GPU. IEEE/ EG Symposium on Volume and Point-Based Graphics, 2008
- [4] Marcus Vesterlund. Simulation and Rendering of a Viscous Fluid using Smoothed Particle Hydrodynamics, 3rd December, 2004
- [5] Matthias Müller, Barbara Solenthaler, Richard Keiser, Markus Gross. Particle-Based Fluid-Fluid Interaction. Eurographics/ACM SIGGRAPH Symposium on Computer Animation (2005), pp. 1–7
- [6] Frank Losasso, Jerry Talton, Nipun Kwatra, Ronald Fedkiw. Two-Way Coupled SPH and Particle Level Set Fluid Simulation. IEEE Transactions on Visualization and Computer Graphics, Issue 4 (July 2008) Volume 14, Pages 797-804, 2008
- [7] Distributed SPH Fluid Simulator (for Hewlett-Packard Scalable Visualization Array). Budapest University of Technology and Economics. Department of Control Engineering and Information Technology. 31p. 2008. <http://www.iit.bme.hu>

\* на процессоре Intel(R) Celeron(R) CPU 2.6GHz 768 Mb RAM

\*\* на процессоре Intel(R) Core(TM)2 E6600 2.4GHZ, 3.25 Gb RAM