

А.А. Бондаренко

*Омский государственный университет им. Ф.М. Достоевского,
г. Омск*

ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМА ОТОБРАЖЕНИЯ ПОВЕРХНОСТИ ЖИДКОСТИ МЕТОДОМ ОБРАТНОЙ ТРАССИРОВКИ ЛУЧЕЙ

При моделировании жидкости со свободной границей модель представляется набором частиц, полученных после применения бессеточных методов SPH, MPS и их модификаций. Данные алгоритмы позволяют рассчитать движение масс жидкости в виде набора взаимодействующих частиц. Для представления результатов моделирования в виде изображения поверхности жидкости необходимо построить и отобразить эту поверхность. Данная работа посвящена разработке приложения, строящего изображение такой поверхности.

Для точного моделирования необходимо как можно больше частиц, представляющих объем жидкости. Увеличение количества частиц приводит к росту затрат на построение изображения. Для повышения производительности вычислений эффективно использовать графический процессор.

Для визуализации поверхности была выбрана техника обратной трассировки лучей. Поверхность жидкости представляется в виде изоповерхности, заданной набором метасфер [1], каждая из которых соответствует отдельной частице в модели жидкости. Таким образом, поверхность определяется уравнением вида $f(\vec{x}) = T$, где T – пороговое значение скалярного поля, $f(\vec{x})$ – функция скалярного поля

$$f(\vec{x}) = \sum_{i=1}^N \varphi_i(|\vec{x} - \vec{p}_i|).$$

Здесь p_i – позиция центра i -й метасферы, N – количество метасфер, $\varphi_i(d)$ – функция плотности i -й метасферы, определяемая выражением

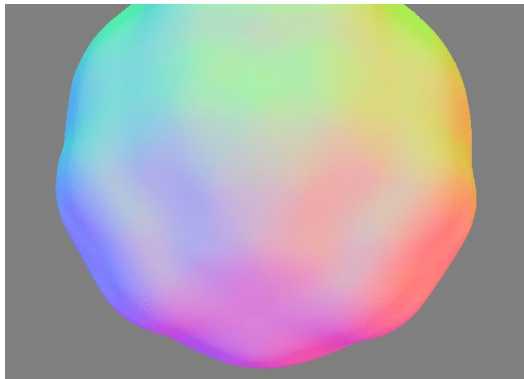
$$\varphi_i(d) = \varphi_0 \left(\frac{d}{r_i} \right),$$

где r_i – радиус метасферы, а $\varphi_0(d)$ – базовая функция плотности. Выбор базовой функции плотности в значительной степени влияет как на качество изображения, так и на производительность, поскольку вычисление её значений происходит многократно в процессе построения изображения.

Простейший алгоритм трассировки лучей для такой поверхности заключается в последовательном расчёте значений скалярного поля f в точках вдоль луча и имеет сложность $O(N \cdot L)$, где N – количество частиц, а L – среднее расстояние от начала луча до точки пересечения луча с поверхностью, либо до границы сцены.

Предлагается алгоритм со сложностью не зависящей от L . В базовой редакции имеет сложность $O(N^2)$, а в оптимизированной – $O(N \cdot \log_2 N)$. Снижение сложности достигается за счёт использования BVH-деревьев [2].

На рис. представлено изображение изоповерхности, полученной с помощью программы, реализующей простейший алгоритм и неоптимизированный вариант альтернативного алгоритма.



Изображения изоповерхности по заданному набору частиц

Литература

1. *Blinn J.F.* A Generalization of Algebraic Surface Drawing // ACM Transactions on Graphics. 1982. № 1 (3). P. 235–256.
2. *Karras T.* Maximizing Parallelism in the Construction of BVHs, Octrees, and k-d Trees. 2012. URL: <https://research.nvidia.com/publication/maximizing-parallelism-construction-bvhs-octrees-and-k-d-trees>.