

Визуализация капель жидкости в FlowVision

А.А. Федоров

ООО "ТЕСИС"

Работа посвящена разработке метода визуализации капель жидкости в программном комплексе FlowVision. FlowVision использует локально адаптивную параллелепипедную сетку с подсеточным разрешением геометрии. Функция VoF используется для восстановления поверхности жидкости и капель, которые затем отрисовываются средствами OpenGL. В работе описываются подходы рендеринга капель жидкости, рассматриваются эффекты преломления и затухания света.

Ключевые слова: визуализация жидкости, рендеринг капель, FlowVision, OpenGL

1. Введение

Инженеры уделяет большое значение аэродинамическим характеристикам аппаратов, проектируемых для применения в воздушной или жидкостной среде. Их оценивают посредством моделирования различных ситуаций в CFD-пакетах¹. Выполненные расчеты позволяют получить количественные характеристики объекта и оценить их соответствие требованиям. Визуализация данных — важная часть оценочного этапа, потому как зрительное восприятие развито у человека больше, чем остальные виды восприятия. Настоящая работа посвящена визуализации капель жидкости в отечественном CFD-пакете FlowVision.

1.1. Обзор существующих решений

Жидкость, как правило, прозрачна, поэтому для реалистичной визуализации необходима работа с полупрозрачными объектами. Одна из идей реализации полупрозрачности тел с преломляющими стенками рассмотрена в работе [3]. Суть её следующая: сначала выполняется отрисовка всех непрозрачных объектов сцены в текстуру фона. Далее рендерятся полупрозрачные объекты, при этом цвет из текстуры фона берётся в смещённом пикселе, смещение регулируется картой нормалей мэша и коэффициентом масштабирования. Результат работы такого алгоритма выглядит правдоподобно, однако его минус в том, что не учитывается расстояние до объектов фона — не наблюдается эффект параллакса.

Более серьёзный подход к реализации эффекта преломления в реальном времени был представлен для капель в работе [4]. Предполагается, что капли имеют форму полусферы и располагаются на экране. Вводится параметр искажения, используемый для преобразования координат пикселя внутри капли степенной функцией (производится нелинейная гомотетия круга, переводящая круг в самого себя). Здесь не наблюдается эффект параллакса, но контроль преломления ближе к физически корректному, чем в подходе [3].

Физически корректная методика симуляции капель жидкости на лобовом стекле автомобиля предложена в работе [5]. При моделировании формы капель помимо гравитации и ветра в расчётах учитывается гистерезис контактного угла и диссипативные силы. Также реализован эффект преломления фона в капле. Помимо этого вычисляются и отрисовываются следы движения капель по стеклу.

Трассировка преломленного луча по карте глубин используется в работе [6] для симуляции эффекта преломления. Этот подход основан на бинарном поиске длины пути луча от места преломления до точки пересечения с фоновой поверхностью, восстанавливаемой по двумерной карте глубин. Он не всегда выдаёт физически корректные точки пересечений, но даёт более реалистичное изображение в реальном времени, чем, например, подход [3].

¹CFD (Computational Fluid Dynamics) - вычислительная гидродинамика - ряд подходов по вычислению характеристик потоков жидкости.

Более того в этом подходе учитывается множественное полное внутреннее отражение.

2. Визуализация капель жидкости

Поскольку система визуализации FlowVision основана на OpenGL, то разработанный метод тоже базируется на OpenGL¹, но в нём не используются какие-либо техники, характерные только для OpenGL, поэтому этот метод может быть реализован с помощью других графических API (например, DirectX). Предварительно стоит учесть, что в FlowVision используется алгоритм обработки полупрозрачных объектов — Depth peeling [7]. Суть его заключается в последовательном получении слоёв геометрической сцены и дальнейшем их слитии.

В FlowVision жидкость представляется в виде функции VoF², заданной на параллелепипедной локально адаптивной сетке с подсеточными разрешениями геометрии поверхностей. Поверхность жидкости восстанавливается алгоритмом, основанным на базе Marching cubes, однако описание этого процесса выходит за рамки статьи. Работа посвящена отрисовке не разрешаемых расчётной сеткой мелких частиц - капель, занимающих долю ячейки. Наличие капли в ячейке определяется значением функции VoF: если оно больше заданной пороговой величины, а в соседних ячейках меньше, — то капля присутствует. Количество капель может достигать десятков и сотен тысяч, что накладывает жёсткие требования на скорость их отрисовки. В то же время необходимо обеспечить приемлемое качество выходного изображения.

Рендеринг большого количества геометрических примитивов упирается в конечную пропускную способность видеокарты. В связи с этим производители графических чипов реализовали возможность генерации геометрии непосредственно на видеокарте³.

Также стоит заранее сказать, что ожидается на выходе: для каждой капли генерируется передняя поверхность сферы⁴. По сравнению с рендерингом полной сферы у этого подхода есть следующие преимущества:

1. меньше количество генерируемых полигонов, и, следовательно, нагрузка на фрагментный шейдер
2. легче рассчитать преломление света внутри частицы

Массив атрибутов, содержащих центры и радиусы капель, отправляется на видеокарту и обрабатывается тривиальным вершинным шейдером. Шейдер контроля тесселяции оценивает уровень детализации частицы исходя из её угловых размеров по отношению к наблюдателю. Далее шейдер оценки тесселяции выполняет генерацию вершин передней полусферы, на основе которых задаётся её аппроксимация треугольниками. Финальный шаг программируемой части конвейера - фрагментный шейдер, рассчитывающий видимый цвет капли.

2.1. Генерация полусфер с помощью тесселяции

2.1.1. Определение детализации

Необходимо обеспечить автоматическое определение уровня детализации капли в зависимости от её угловых размеров относительно наблюдателя.

Пусть на входе дано следующее:

¹Используется OpenGL 4.0 compatibility profile, см. спецификацию [8].

²VoF — Volume of Fluid, доля объема ячейки, занимаемая жидкостью. Подробнее см. [9].

³В графическом конвейере за это отвечают стадии *тесселяции*. Подробнее см. [8, 10].

⁴Под передней поверхностью сферы имеется в виду полусфера, которую бы увидел наблюдатель в ортогографической проекции, если бы сфера была непрозрачной

R радиус сферы
 d расстояние от наблюдателя до центра сферы
 $w \times h$ разрешение финального изображения (в пикселях)
 Δp допустимая погрешность аппроксимации (в пикселях)

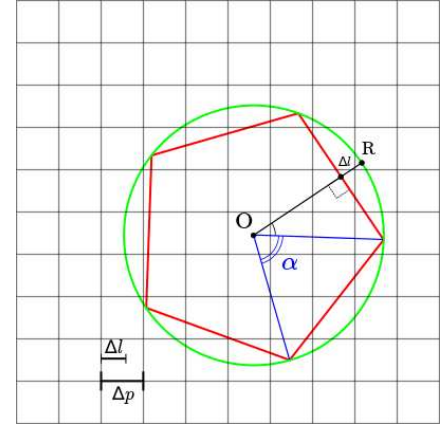
Требуется оценить α – угол сектора внешней окружности, разделяющей переднюю и заднюю полусферы.

Максимальное отклонение аппроксимации внешней окружности от аналитической формы выражается так (см. рис. 1)

$$\Delta l = R \cdot \left(1 - \cos \frac{\alpha}{2}\right)$$

Пусть задана матрица перспективной проекции:

$$P = \begin{pmatrix} s_x & 0 & 0 & d_x \\ 0 & s_y & 0 & d_y \\ 0 & 0 & s_z & d_z \\ 0 & 0 & -1 & 0 \end{pmatrix}$$



Тогда можно оценить сверху отклонение в пространстве NDC: 1: Определение детализации сферы

$$\Delta l_{ndc} = \Delta l \cdot \frac{\max(s_x, s_y)}{d}$$

Назначение количества различаемых пикселей заключается в следующем неравенстве:

$$\frac{\Delta l_{ndc}}{2} \cdot \max(w, h) \leq \Delta p$$

Отсюда выводится оценка сверху на размер сектора²:

$$\alpha \geq \alpha_0 = 2 \arccos \left(1 - \frac{2\Delta p \cdot d}{R \cdot \max(s_x, s_y) \cdot \max(w, h)} \right)$$

Количество секторов на внешней окружности: $N = \left\lceil \frac{2\pi}{\alpha_0} \right\rceil$. Это значение будет использоваться для определения детализации полусферы.

2.1.2. Получение треугольников

Используется режим тесселяции *quads, equal_spacing*: на вход поступают равномерно распределённые абстрактные координаты $(u; v)$ из диапазона $[0; 1]$ (см. рис. 2).

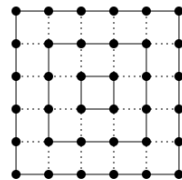


Рис. 2: Разбиение абстрактного квадрата

¹NDC (Normalized Device Coordinates) - пространство, в котором трехмерные координаты, лежат в диапазоне $[-1; 1]$. Эти координаты в дальнейшем переводятся в пространство экрана.

²Для случая ортогографической проекции нужно положить $d = 1$, тем самым исключив перспективное деление. В остальном все расчёты остаются такими же.

$$\begin{aligned} (u'; v') &= (2u - 1, 2v - 1) \\ r_{part} &= 1 / \max(u', v') \\ \begin{pmatrix} x \\ y \\ z \end{pmatrix} &= R \cdot \begin{pmatrix} r_{part} \cdot \frac{u'}{\sqrt{u'^2 + v'^2}} \\ r_{part} \cdot \frac{v'}{\sqrt{u'^2 + v'^2}} \\ \sqrt{1 - r_{part}^2} \end{pmatrix} \end{aligned}$$

На выходе $(x; y; z)$ - смещение вершины относительно центра сферы в пространстве наблюдателя. Иллюстрация работы шейдеров тесселяции представлена на рис. 3.



Рис. 3: Триангуляция, генерируемая шейдерами тесселяции

Наблюдается увеличение секторной детализации аппроксимаций окружностей от центра к краю. Итоговое разбиение сферы выглядит равномерным - проецируемые на экран треугольники имеют почти одинаковые площади. Это нужно для поддержания единой точности интерполяции вершинных атрибутов (здесь - нормалей).

2.2. Расчёт цвета в фрагментном шейдере

2.2.1. Преломление луча взгляда

Как было заявлено ранее, необходимо рассчитывать цвет не только передней поверхности сферы, но и учитывать заднюю часть. Точку выхода луча из сферы (она же точка задней поверхности) несложно определить, руководствуясь законом Снеллиуса.

2.2.2. Модель освещения

Рассмотрим освещение отдельной капли источником направленного света. Для начала нужно договориться о модели освещения. Будем рассчитывать диффузную компоненту освещения с помощью модели Ламберта, а бликовую - Кука-Торренса¹.

2.2.3. Преломление параллельных лучей света

Как передний, так и задний пиксель могут быть освещены либо прямым лучом света $\mathbf{L}$, либо преломленным в шаре. Установить это можно по знаку скалярного произведения луча света на внешнюю нормаль сферы. Например, для передней точки с нормалью $\mathbf{n}_f$ если $\text{dot}(\mathbf{L}, \mathbf{n}_f) < 0$, то точка освещена прямым светом. В противном случае - преломленным.

В случае преломленного освещения необходимо найти точку входа луча света. Пусть:

Угол α поможет определить внешнюю нормаль в точке входа, а следовательно, и саму точку входа света. По этим данным можно будет найти преломленный луч света (см. рис. 4).

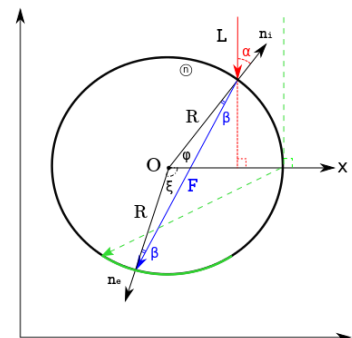


Рис. 4: Преломление луча света на частице

¹Подробно модель Кука-Торренса изложена в работе [11]

- L** направление распространения луча света
- x** направление, перпендекулярное **L** и такое, что $(\mathbf{x}, \mathbf{n}_i) > 0$
- α угол между **L** и внешней нормалью в точке входа $\mathbf{n}_i$
- β угол между преломленным лучом света **F** и $\mathbf{n}_i$
- ξ угол между направлением **x** и внешней нормалью в точке выхода $\mathbf{n}_e$
- φ угол между внешней $\mathbf{n}_i$ и **x**

Произведя ряд математических операций, получаем уравнение четвёртой степени на $t = \sin \beta$:

$$f(t) := 4 \cdot t^4 - 4n \cos \xi \cdot t^3 + (n^2 - 4) \cdot t^2 + 2n \cos \xi \cdot t + \cos^2 \xi = 0 \quad (1)$$

Заметим, что $\cos \xi$ нам на самом деле известно:

$$\cos\left(\frac{\pi}{2} - \xi\right) = (\mathbf{L}, \mathbf{n}_e) \Rightarrow \cos \xi = \pm ||[\mathbf{L} \times \mathbf{n}_e]||$$

Решать уравнение (1) будем методом Ньютона. Из цепочки равенств $nt = n \sin \beta = \sin \alpha \leq 1$ получаем область поиска $t \in [0; \frac{1}{n}]$. Исследуем количество решений, попадающих в указанных диапазон. Построим график зависимости $\xi(\varphi)$ для нескольких значений показателя преломления n (рис. 5).

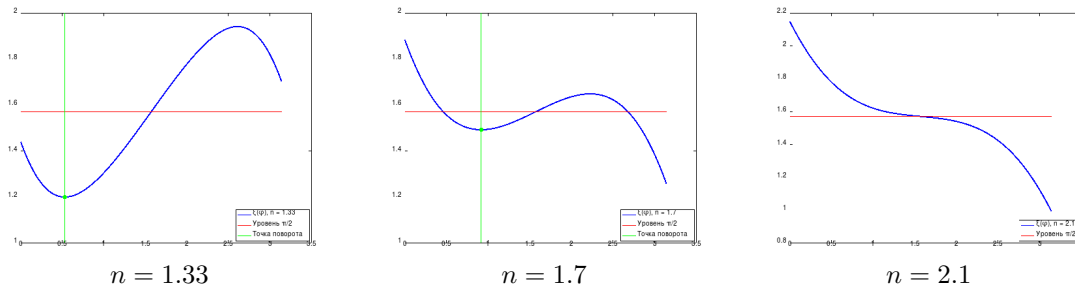


Рис. 5: $\xi(\varphi) = \pi - \varphi - 2 \arcsin \frac{\cos \varphi}{n}$

По факту уравнение (1) является уравнением на $\frac{\cos \varphi}{n}$, а значит, и на φ (поскольку $\varphi \in [0; \pi]$). То есть ставится задача обращения функции $\xi(\varphi)$. Точки поворота¹ этой функции лежат во множестве решений уравнения (2):

$$\frac{d\xi}{d\varphi} = -1 + \frac{2 \sin \varphi}{\sqrt{n^2 - \cos^2 \varphi}} = 0 \quad (2)$$

Решения уравнения (2) в диапазоне $[0; \pi]$ следующие:

$$\begin{aligned} \varphi_1 &= \arcsin \frac{\sqrt{n^2 - 1}}{\sqrt{3}} \\ \varphi_2 &= \pi - \varphi_1 \end{aligned}$$

Для показателя преломления $n > 2$ функция $\xi(\varphi)$ не имеет точек поворота.

Для корректной работы метода Ньютона необходимо произвести разделение области определения функции $\xi(\varphi)$ на подобласти, в которых функция обращается однозначно. Опустим технические подробности этого процесса.

На данной задаче метод хорошо сходится уже на 2-3 итерации. Зная φ несложно восстановить нормаль в точке входа и, соответственно, найти саму точку входа.

¹точки поворота - такие значения φ^* , что $\frac{d\xi}{d\varphi}$ меняет знак при переходе через φ^*

2.2.4. Формулы Френеля

Будем считать свет неполяризованным. Неполяризованный свет можно представить как смесь света с s -поляризацией и p -поляризацией с равными вкладами. Вычислим коэффициент прохождения света через одну поверхность сферы (T - для луча взгляда, T_f - для преломленного луча света, попадающего на точку передней поверхности, T_b - для преломленного луча света, попадающего на точку задней поверхности), руководствуясь [12].

2.2.5. Формулы Френеля и направленный свет

Стоит подробно остановиться на применении коэффициентов Френеля к освещению источником направленного света. Выбранная нами модель освещения предполагает наличие прямого отражённого от освещаемой поверхности луча (бликовая компонента) и сферической волны (диффузная компонента). Формулы для коэффициентов представлены в табл. 1, а их обоснование приведено ниже.

Поверхность	Попадание луча	Диффузный фактор	Бликовый фактор
Передняя	Прямое	1	$1 - T_f$
Передняя	Преломленное	T_f	$T_f \cdot T_f$
Задняя	Прямое	T	$T_b \cdot T$
Задняя	Преломленное	$T_b \cdot T$	$T_b \cdot (1 - T_b) \cdot T$

Таблица 1: Учёт коэффициентов Френеля при освещении частицы направленным светом

В случае освещения передней поверхности прямым светом фактор диффузной компоненты разумно задать единичным, а бликовой - как коэффициент отражения при преломлении луча света на передней поверхности сферы.

Для случая освещения передней поверхности сферы преломленным светом фактор диффузной компоненты имеет смысл задать как T_f , учтя таким образом прохождение света через заднюю поверхность частицы, а фактор бликовой компоненты - T_f^2 , учтя, вдобавок, выход луча из частицы (предполагается, что наблюдатель находится снаружи).

Когда речь идет об освещении задней поверхности, то разумно предварительно домножать как диффузный, так и бликовый факторы на коэффициент прохождения луча взгляда T через переднюю поверхность сферы. Бликовую компоненту разумно также домножить на коэффициент прохождения луча света сквозь заднюю поверхность сферы.

Рассмотрим теперь коэффициенты в случае попадания преломленного луча света на заднюю поверхность шара. Диффузный фактор: свет сначала входит в частицу (домножение на T_b), потом рассеивается на задней поверхности и один из лучей попадает в наблюдателя - отсюда домножение на T (по аналогии с предыдущим случаем). Бликовый фактор: прохождение через поверхность сферы (T_b), отражение от задней поверхности ($1 - T_b$) и попадание в наблюдателя (T).

Таким образом были определены правила корректировки интенсивности света в различных типовых случаях следования луча света.

2.2.6. Поглощение света

Согласно закону Бугера¹, интенсивность света, прошедшего через прозрачную среду, изменяется следующим образом:

$$I = I_0 \cdot e^{-\chi l}$$

¹см. [12], глава VIII, §84

Примем на вооружение эту формулу и добавим коэффициент поглощения χ в список входных параметров. Применять её будем к задней поверхности сферы и преломленным лучам источника направленного света.

2.3. Удаление невидимых частиц

Часть частиц может быть скрыта за непрозрачными объектами сцены. Детектирование таких частиц и исключение их из общего массива, отправляемого для отрисовки, может увеличить производительность рендеринга. Для этой цели используется техника Transform Feedback, позволяющая захватывать массивы атрибутов с видеокарты. Создаются специальные шейдеры, проверяющие, перекрывается ли частица фоном или нет.

Проверка видимости частицы происходит в геометрическом шейдере, который определяет координаты ближайшей к наблюдателю точки сферы в экранном пространстве и сравнивает значение её глубины с значением глубины из z-буфера фона (предварительно отрендеренной непрозрачной части сцены). Если проверка проходит успешно (точка имеет меньшую глубину), то считается, что частица видна и её атрибуты записываются в выходной поток.

Обычно в таких случаях используется методика Occlusion Culling, основывающаяся на вычислении MIP-уровней карты высот и взятии глубины с такого MIP-уровня, чья детализация грубее размеров проекции габаритного параллелепипеда объекта на экран. В алгоритме Depth Peeling карт глубин столько же, сколько слоёв, и вычисление их MIP-уровней может повлиять на производительность. Поэтому было принято решение о релализации описанной выше проверки в ущерб корректности - капли, пересекающиеся с поверхностями могут исчезать и появляться в зависимости от ракурса.

Выходной поток геометрического шейдера захватывается с помощью Transform Feedback. В результате получается массив такого же формата, который изначально подавался в вершинный шейдер, но меньшего размера - невидимые частицы были отброшены. Далее этот массив отдаётся основной цепочке шейдеров, описанной в предыдущих разделах.

Удаление невидимых частиц можно производить каскадно: перед отрисовкой очередного полупрозрачного слоя отсечём те частицы, у которых ближайшая к наблюдателю точка имеет глубину больше, чем соответствующий ей пиксель предыдущего слоя. Более того, во время первого отсечения можно оценить уровень детализации частицы и отсечь слишком маленькие.

3. Результаты

Была разработана и реализована методика визуализации капель жидкости для программного комплекса FlowVision. Эта методика была опробована на задаче качения шины. До внедрения этой методики капли визуализировались параллелепипедами сплошного цвета и "тонкими" в плане полупрозрачными стенками. Ниже приведены изображения, демонстрирующие результаты работы применённых методов.

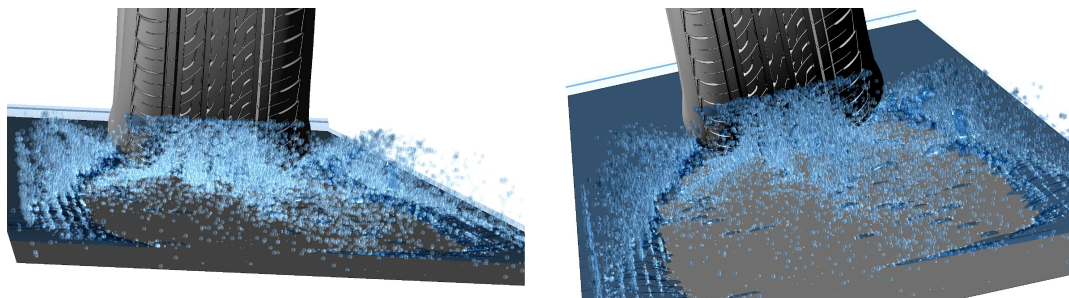


Рис. 6: Новый метод визуализации капель

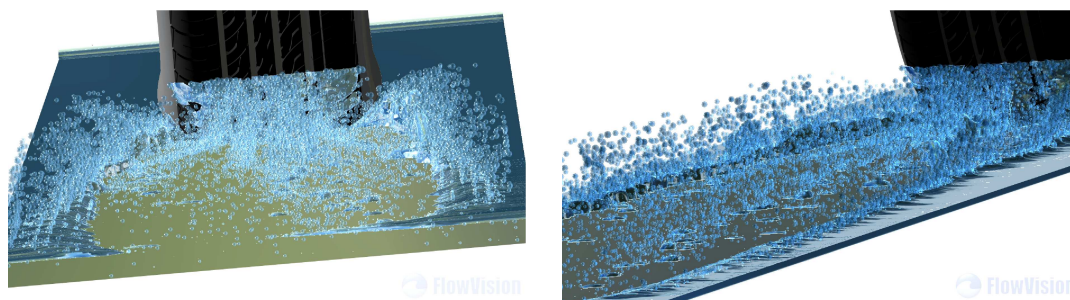


Рис. 7: Новый метод визуализации капель

4. Заключение

В результате выполненной работы была усовершенствована визуализация капель жидкости в программном комплексе FlowVision.

Разработан алгоритм рендеринга сферических частиц с генерацией динамически детализированной геометрии на видеокарте и эффектами объёмного поглощения и преломления направленного света, отражения и преломления *subtmap*-текстуры с учётом коэффициентов Френеля. Также реализовано каскадное отсечение невидимых и уже отрисованных частиц для повышения скорости рендеринга в режиме полупрозрачности Depth Peeling.

Литература

1. Matthias Müller-Fischer. Fast Water Simulation for Games Using Height Fields. URL: <http://matthias-mueller-fischer.ch/talks/GDC2008.pdf> (дата обращения 01.04.2017)
2. Yury Kryachko. Using Vertex Texture Displacement for Realistic Water Rendering. URL: http://http.developer.nvidia.com/GPUGems2/gpugems2_chapter18.html (дата обращения 27.02.2017).
3. Tiago Sousa. Generic Refraction Simulation. URL: http://http.developer.nvidia.com/GPUGems2/gpugems2_chapter19.html (дата обращения 27.02.2017).
4. Yonggao Yang, Changqian Zhu and Hua Zhang. Real-time simulation: water droplets on glass windows // Computing in Science & Engineering, IEEE, vol. 6, issue 4, July-Aug. 2004, pp. 69-73, DOI: 10.1109/MCSE.2004.20
5. Nobuyuki Nikata, Masanori Kakimoto, Tomoyuki Nishita. Animation of Water Droplets on a Hydrophobic Windshield. URL: https://www.researchgate.net/publication/268438621_Animation_of_Water_Droplets_on_a_Hydrophobic_Windshield (дата обращения 02.03.2017).
6. Scott T Davis, Chris Wyman. Interactive refractions with total internal reflection // Proceedings of Graphics Interface 2007, pp. 185-190, Montreal, Canada.
7. Cass Everitt. Order-Independent Transparency. URL: <http://www.eng.utah.edu/~cs5610/lectures/OrderIndependentTransparency.pdf> (дата обращения 23.03.2017).
8. Mark Segal, Kurt Akeley. The OpenGL Graphics System: A specification (Version 4.0(Compatibility Profile) - March 11, 2010). URL: <https://khronos.org/registry/OpenGL/specs/gl/glspec40.compatibility.pdf> (дата обращения 02.04.2017).

9. Голов Андрей Владимирович. Моделирование движения многофазной жидкости в программном комплексе FlowVision, URL:
https://mipt.ru/education/chair/computational_mathematics/upload/45a/2013_ApplMath_MSc_731_Golov-arphj5jlf2t.pdf (дата обращения 06.03.2017).
 10. Боресков Алексей Викторович. Тесселяция в современном OpenGL. URL:
<http://steps3d.narod.ru/tutorials/tesselation-tutorial.html> (дата обращения 27.02.2017).
 11. Robert L. Cook, Kenneth E. Torrance. A Reflectance Model for Computer Graphics. *Computer Graphics*, vol. 15, no. 3, 1981, ACM.
 12. Сивухин Дмитрий Владимирович. Общий курс физики. // Учеб. пособие: Для вузов. В 5 т. Т.IV. Оптика. – 3-е изд. стереот. - М.: ФИЗМАТЛИТ, 2005
-

Droplet visualization in FlowVision

Fedorov A.A.

TESIS Ltd.

This paper is devoted to developing a method for droplet visualization in FlowVision. FlowVision uses local adaptive parallelepiped computational grid with subgrid geometry resolution. The VOF function is used to reconstruct fluid surface and droplets which are then rendered using OpenGL. Approaches to improved rendering droplets are considered in this paper including refraction and lighting effects.

Keywords: fluid rendering, droplet rendering, FlowVision, OpenGL

References

1. Müller-Fischer M.. Fast Water Simulation for Games Using Height Fields. URL: <http://matthias-mueller-fischer.ch/talks/GDC2008.pdf> (accessed 01.04.2017)
2. Kryachko Y.. Using Vertex Texture Displacement for Realistic Water Rendering. URL: http://http.developer.nvidia.com/GPUGems2/gpugems2_chapter18.html (accessed 27.02.2017).
3. Sousa T.. Generic Refraction Simulation. URL: http://http.developer.nvidia.com/GPUGems2/gpugems2_chapter19.html (accessed 27.02.2017).
4. Yang Y., Zhu C. and Zhang H.. Real-time simulation: water droplets on glass windows // Computing in Science & Engineering, IEEE, vol. 6, issue 4, July-Aug. 2004, pp. 69-73, DOI: 10.1109/MCSE.2004.20
5. Nobuyuki Nikata, Masanori Kakimoto, Tomoyuki Nishita. Animation of Water Droplets on a Hydrophobic Windshield. URL: https://www.researchgate.net/publication/268438621_Animation_of_Water_Droplets_on_a_Hydrophobic_Windshield (accessed 02.03.2017).
6. Davis S.T., Wyman C.. Interactive refractions with total internal reflection // Proceedings of Graphics Interface 2007, pp. 185-190, Montreal, Canada.
7. Everitt C.. Order-Independent Transparency. URL: <http://www.eng.utah.edu/~cs5610/lectures/OrderIndependentTransparency.pdf> (accessed 23.03.2017).
8. Segal M., Akelet K. The OpenGL Graphics System: A specification (Version 4.0(Compatibility Profile) - March 11, 2010). URL: <https://khronos.org/registry/OpenGL/specs/gl/glspec40.compatibility.pdf> (accessed 02.04.2017).
9. Golov A.V. *Modelirovanie dvizheniya mnogofaznoy zhidkosti v programmnom komplekse FlowVision* [Modelling fluid motion in Flowvision package]. URL: https://mipt.ru/education/chair/computational_mathematics/upload/45a/2013_ApplMath_MSc_731_Golov-arphj5jlf2t.pdf (accessed 06.03.2017).
10. Boreskov A.V. *Tesselyatsiya v sovremennom OpenGL* [Tessellation in modern OpenGL]. URL: <http://steps3d.narod.ru/tutorials/tessellation-tutorial.html> (accessed 27.02.2017).
11. Cook R.L., Torrance K.E. A Reflectance Model for Computer Graphics. *Computer Graphics*, Vol. 15, No. 3, 1981, ACM.

12. Sivuhin D.V. *Obzchiy kurs phisiki* [General course of physics] // Textbook: for universities. Vol.IV. Optics. – edition 3 stereotype - M.: FIZMATLIT, 2005.