А.А. Колпаков
Муромский институт (филиал) Владимирского государственного университета
602264 г. Муром, Владимирской обл., ул. Орловская, 23
E-mail: desT.087@gmail.com

Алгоритм аудио микширования для использования в системах с массово-параллельной архитектурой

Хотя графические процессоры достаточно производительны, существует несколько проблем в использовании их для аудио микширования, которые связаны с архитектурой и ограниченностью функционала.

Во-первых, пропускная способность шины между графическим процессором и основной памятью, которая меньше, чем между основным процессором и основной памятью. Поскольку общие вычисления на GPU базируются на 3D рендеринге, скорость записи обычно выше, чем скорость чтения. Такая асимметрия делает процедуру считывания результата достаточно продолжительной [1].

Во-вторых, общие вычисления на GPU базируются на 3D моделях, различные задачи требуют различных настроек GPU, таких как 3D модели, трансформирующие матрицы и программы шейдеров. Также CPU вынужден периодически проверять статус GPU.

В-третьих, недостатком GPU является производительность в логических операциях. Как известно, CPU отслеживает ветвления, GPU же работает по другому: каждая ветка ветвления сначала выполняется, а потом уже выбирается нужный результат. Это делает распараллеливание легче, но требует больше ресурсов.

И наконец, набор инструкций GPU несовместим с набором CPU. Кроме того, время выполнения и длина кода лимитированы. Все это делает сложным перенос существующих алгоритмов на GPU.

В разрабатываемом алгоритме предлагается загрузка единой текстуры. В качестве примера представлен первый проход алгоритма. Входные данные содержат n семплов последовательностей размером L байт и n коэффициентов затухания. Используется два текстурных буфера формата RGBA. Текстура T1 имеет размерность [L/4]*n и хранит все семплы последовательностей в линию. Текстура T2 имеет размерность 1*n и хранит коэффициенты затухания. Координаты всех текстур приведены в таблице 1.

Таблица 1. Координаты текстур

Вершины	Координаты текстуры T1	Координаты текстуры T1
(-1, -1, 0)	(1/2w, 1)	(0.5, 1)
(1-2/(w+1), -1, 0)	(1, 1)	(0.5, 1)
(-1, 1, 0)	(1/2w, 0)	(0.5, 0)
(1-2/(w+1), 1, 0)	(1, 0)	(0.5, 0)

Аудио микширование производится независимо для каждого пикселя. Процесс микширования для пикселя с координатам (x,y) приведено в виде псевдокода, показанного ниже

```
float2 ptCur=pt.tO ;
for(int y=0 ; y<TOTAL_SOURCE ; y++)
if(y!=pt.v.Y)
{
ptCur.y=(Y+0.5)/TOTAL_SOURCE;
iSum + = decodePCMU(tex2D(s[0], ptCur)*256);
}
int4 w = tex2D(s[1], pt.t1)*256 ;
iSum=iSum * w.b/16;
```

Текстурные координаты пикселя (x,y) рассчитываются путем интерполяции текстурных координат вершин. Исходя из таблицы 1, текстурная координата для T1 или pt.t0 должна быть (X,Y), а для T2 или pt.t1 – (0.5,Y). Pt.t0 указывает на семпл, для которого производятся текущие преобразования микширования, данный семпл назовем «точкой прицеливания». Для исключения появления эха другая текстурная координата ptCur ограничена по доступу T1 вместо pt.t0.

Составляющая x текстуры ptCur идентична текстуре pt.t0, а составляющая y рассчитывается из циклической переменной, которая обозначает каждый источник звука. В цикле регистр положения v используется, чтобы пропустить «точку прицеливания». Наконец, коэффициенты затухания считываются из текстуры pt.t1. Поскольку коэффициент хранится в первом байте, выборка производится только по синей составляющей.

Тестовые входные данные для микширования представляют собой 8 последовательностей по 320 семплов. Все семплы сгенерированы случайно. В качестве тестового стенда использован компьютер с процессором Intel Core i3-4130, 4 ГБ оперативной памяти, графическая карта NVIDIA GeForce GT730. Результаты для CPU и GPU на выходе идентичны. Время работы алгоритма и дисперсия результатов приведены в таблице 2.

Таблица 2. Результаты тестирования

Вычислительное устройство	Время работы алгоритма, мс	Дисперсия результатов
CPU	1.351	3.757
GPU	2.226×10^{-1}	1.426×10^{-3}

Результаты тестирования показывают, что GPU требуется в шесть раз меньше времени на обработку данных, чем CPU. Однако GPU использует только 44.1% времени на вычисления, остальное тратит на операции ввода/вывода, тогда как CPU использует 99.8% времени на вычисления.

В данной работе представлен алгоритм аудио микширования для использования вычислений общего назначения на графических процессорах. Главное его достоинство – это комбинирование множества этапов микширования путем использования двухпроходного рендера, что существенно снижает время переключения между буферами. Использование для расчетов одной текстуры повышает эффективность операций ввода/вывода. Хотя операции ввода/вывода занимают приблизительно половину времени вычислений, данный алгоритм достаточно производителен и пригоден для практического использования.

Литература

1. Колпаков А.А. Теоретическая оценка роста производительности вычислительной системы при использовании нескольких вычислительных устройств / Колпаков А.А. // В мире научных открытий, 2012. – №1. – С. 206-209.

2. Кропотов Ю.А., Ермолаев В.А. О корреляционом оценивании параметров модели акустических эхо-сигналов / Кропотов Ю.А., Ермолаев В.А.. // Вопросы радиоэлектроники, 2010. – №1. – С. 46-50.

3. Kropotov Y.A., Ermolaev V.A. Investigation of model parameters of acoustic echo correlation estimation method Proceedings of 20-th International Crimean Conference "Microwave & Telecommunication Technology" (CriMiCo’2010). Sevastopol, 2010, v.1, p. 422 - 423.