

разработке современных игр и интерактивных веб-приложений. API достаточно высокоуровневый, продуман до мелочей, самодостаточен, легок в освоении и особенно элегантно интегрируется в приложения, использующие WebGI и WebRTC.

Современные системы поиска аудиофайлов часто руководствуются такими критериями, как название, исполнитель, текст песни, однако они осуществляют лишь однозначный поиск музыкальной композиции, и осуществление поиска схожих по определенным параметрам аудиозаписей не представляется возможным. Более универсальным и более интеллектуальным является проведение поиска по составлению прогноза о том, какие аудиозаписи слушают другие пользователи с такой же начальной выборкой, как у субъекта поиска. Однако такой подход также может оказаться не очень полезен, если пользователь в качестве примера предоставил непопулярную аудиозапись, которую тяжело найти у других пользователей. Самые хорошие результаты в поиске похожих аудиофайлов будут давать методы, основанные только на анализе звучания предоставленного пользователем файла, не основываясь на результатах других пользователей.

Существует несколько подходов к анализу звучания. Все они сводятся к выявлению характеристик аудиозаписи и сравнению их с характеристиками остальных аудиозаписей из какого-либо источника данных, например, БД. Отличаются в этих подходах лишь методы выявления характеристик и методы сравнения аудиозаписей.

Для выявления характеристик аудиозаписи распространен метод сравнения спектрограмм. Сначала с помощью преобразования Фурье строится спектрограмма аудиозаписи - зависимость спектральной плотности мощности сигнала от времени. Преобразование Фурье выполняет задачу сопоставления функции вещественной переменной другой функции вещественной переменной, которая описывает коэффициенты («амплитуды») при разложении исходной функции на элементарные составляющие - гармонические колебания. Далее происходит выделение стойких к помехам характеристик звукозаписи, которые и будут участвовать в сравнении.

Существует много реализаций данного метода, но все имеют существенный недостаток - заточены на работу с определенным форматом данных. В данной научной работе предлагается унифицированный подход, заключающийся в представлении аудиофайлов в формате `aggaubuffer`, так как все известные форматы аудиофайлов представимы в нем.

Сравнивать аудиофайлы можно либо используя метрики определения кратчайшего расстояния, либо используя сравнение с шаблоном, в качестве которого может выступать "показательный" аудиофайл для какой-либо группы аудиозаписей. Второй метод требует помимо наличия выборки аудиофайлов еще и генерации шаблонов, что затратно и потому не оптимально. Первый метод более предпочтителен и именно он будет использоваться в научной работе.

Таким образом, составление рекомендованных к прослушиванию аудиозаписей будет производиться исходя из анализа звучания начальной выборки, выделением из спектрограммы устойчивых к помехам амплитудных и частотных значений сигнала, и последующим определением аудиофайлов, чьи характеристики имеют кратчайшее расстояние до характеристик анализируемых аудиофайлов.

Исследование поддержано проектом CERES. Centers of Excellence for young REsearchers (Reg.no. 544137-TEMPUS-1-2013-SK-JPHES),



Список использованных источников:

1. Гольденберг Л. М. Цифровая обработка сигналов.
2. Афонский А. А., Дьяконов В. П. Цифровые анализаторы спектра, сигналов и логики.
3. Web Audio API – новые возможности генерации, обработки и объемного распределения звука в браузере. [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://html5.by/blog/audio/>.

АНАЛИЗ АЛГОРИТМА БЫСТРОГО ОБРАТНОГО КВАДРАТНОГО КОРНЯ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Фокин Н.В.

Искра Н.А. – ст. преподаватель каф. ЭВМ

Во время разработки коммерческого программного обеспечения наступает такой момент, когда вычислительной мощности большинства персональных систем становится недостаточно. Разработчик становится перед выбором: ждать появления на рынке новых технологий, а также жертвовать частью рынка, или использовать различные методы оптимизации. В свое время для повышения производительности CPU

применялись разные алгоритмы. Одним из них и является алгоритм быстрого обратного корня.

В данном алгоритме нет медленных операций деления и квадратного корня, вместо этого используются операции вычитания, битового сдвига, а также дробное вычитание и умножение. Точности (менее 0.2% в меньшую сторону и никогда – в большую) полученных результатов достаточно для трехмерной графики, но не хватает для настоящих численных расчетов.

Данный алгоритм был разработан в Silicon Graphics, а реализация появилась в 1999 году в исходном коде компьютерной игры Quake 3 Arena, где он использовался для ускорения обработки графики вычислительными блоками, а именно для вычисления нормализованного вектора. Так как программы с 3D-графикой используют данные нормализованные векторы для определения освещения и отражения, за секунду должны выделяться миллионы данных корней.

С математической точки зрения в основе данного алгоритма лежат две операции:

Первой важной операцией является аппроксимация для нахождения первого приближенного значения. Любую функцию $y = f(x)$ можно представить как сумму: $y = a_0 + a_1x + \dots + a_nx^n$, где a_0, a_1, a_n – константы. Для квадратного корня сумма имеет бесконечное число аргументов. Тогда, остановившись на некотором x_n можно получить результат с некоторой точностью. Для обеспечения более быстрой работы алгоритма используются только два аргумента. Формула имеет вид: $y = a_0 + a_1x$. Эмпирически [1, 2] было получено, что наиболее точный результат дают константы: $a_0 = 0x5f3759df$, $a_1 = -0.5$.

В данном случае полученной точности недостаточно и для уточнения результата используется одна итерация метода Ньютона. Для этого необходимо найти $x_0 = \frac{1}{\sqrt{x}}$

После возведения в квадрат и переноса аргументов в левую сторону имеем функцию: $x_0^{-2} - x = 0$. Тогда, взяв производную и подставив все значения для нахождения x_{n-1} , по методу Ньютона получим:

$$x_{n-1} = x_n - \frac{x_0^{-2} - x}{-2x_0^{-3}} = \frac{3}{2}x_n - \frac{x}{2}x_0^3$$

Основной проблемой данного алгоритма является то, что для чисел типа double алгоритм не дает выигрыша точности, а также контрпродуктивен при работе на некоторых современных процессорах, что было подтверждено на практике.

Таким образом, главная задача состоит в реализации алгоритма для современных процессоров. Данный алгоритм позволит оптимизировать в современных программах с 3D-графикой операции, использующие нормализованные векторы.

Список использованных источников:

1. С. Lomont. Fast inverse square root. Technical Report. 2003.
2. Christian Plesner Hansel. 0x5f3758df, further investigations into accuracy and generalizability of the algorithm. 2012.

МЕТЕОСТАНЦИЯ НА БАЗЕ КОНТРОЛЛЕРА ARDUINO UNO

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Хамутовский Я. И.

Луцук Ю.А. – к.т.н., доцент

Микроконтроллеры позволяют буквально оживить компьютерный код и создавать устройства, которые способны выполнять различные поставленные задачи. На базе микроконтроллеров реализуются различные системы для контроля, управления и обработки данных.

В современном мире мы часто сталкиваемся с электронными термометрами, но никогда не задумываемся о том, как они работают. Иногда необходимо иметь несколько термометров для измерения широкого диапазона температур. Именно по этой причине было решено создать метеостанцию, которая бы имела преимущество в ширине измерения диапазона температур, в отличие от готовых решений рынка.

В данной работе мы создадим метеостанцию, используя микроконтроллер Atmega и термоэлектрический преобразователь. Термоэлектрические преобразователи – иначе, термопары, действуют по принципу термоэлектрического эффекта, то есть благодаря тому, что в любом замкнутом контуре (из двух разнородных полупроводников или проводников) возникнет электрический ток, в случае если места спаев отличаются по температуре (Рис 1). [1]