

ПОВЫШЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ АЛГОРИТМА КАННИ С ПОМОЩЬЮ РЕАЛИЗАЦИИ ПОИСКА ГРАДИЕНТА НА ГРАФИЧЕСКИХ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОРАХ NVIDIA

Балашова О.В.

Научный руководитель: Аксенов С.В.
Томский политехнический университет
634050, Россия, г. Томск, пр-т Ленина, 30
E-mail: balashova@tpu.ru

Введение

В настоящее время постоянно растущий объем графической информации требует специальных методов для ее анализа и обработки. Мощным устройством, которое может быть использовано при решении широкого класса вычислительных задач является графический процессор (GPU – Graphics Processing Unit). Современные GPU представляют собой массивно-параллельные вычислительные устройства, обладающие высокой степенью быстродействия и большим объемом собственной памяти. В данной работе рассматривается технология программирования CUDA. Целью работы является осуществления поиска градиента изображения, реализованного на GPU.

Технология CUDA

Технология CUDA (Compute Unified Device Architecture) была предложена компанией NVIDIA с целью облегчения написания GPGPU – приложений. Данная технология предназначена для разработки приложений для массивно-параллельных вычислительных устройств. Основными ее преимуществами является простота (все программы пишутся на расширенном языке C), набор готовых инструментов и библиотек, кроссплатформенность. Концепция CUDA заключается в следующем: GPU выступает в роли массивно-параллельного сопроцессора к CPU. Программа на CUDA задействует как CPU так и GPU. При этом последовательная часть кода выполняется на CPU, а для массивно – параллельные вычисления осуществляются на GPU, представляющим набор одновременно выполняющихся потоков.

Алгоритм Канни

Детектор ребер Канни предназначен для поиска границ объектов на изображении. **Края**(границы) – это такие кривые на изображении, вдоль которых происходит резкое изменение яркости или других видов неоднородностей.

Алгоритм состоит из пяти отдельных шагов:

1. **Сглаживание.** Размытие изображения для удаления шума. Удаление шума на изображении происходит посредством применения фильтра Гаусса

2. **Поиск градиентов.** Границы отмечаются там, где градиент изображения приобретает максимальное значение. На данном этапе применяется оператор Собеля, дискретный дифференциальный оператор, вычисляющий приближенное зна-

чение градиента яркости изображения. Оператор Собеля основан на свёртке изображения небольшими целочисленными фильтрами в вертикальном и горизонтальном направлениях. Результатом применения оператора Собеля в каждой точке изображения является либо вектор градиента яркости в этой точке, либо его норма.

3. **Подавление не-максимумов.** Только локальные максимумы отмечаются как границы. Пикселями границ объявляются пиксели, в которых достигается локальный максимум градиента в направлении вектора градиента. Значение направления должно быть кратно 45° .

4. **Двойная пороговая фильтрация.** Потенциальные границы определяются порогами. Выделение границ Канни использует два порога фильтрации: если значение пикселя выше верхней границы – он принимает максимальное значение (граница считается достоверной), если ниже – пиксель подавляется, точки со значением, попадающим в диапазон между порогов, принимают фиксированное среднее значение (они будут уточнены на следующем этапе).

5. **Трассировка области неоднозначности.** Итоговые границы определяются путём подавления всех краёв, несвязанных с определенными (сильными) границами. Задача сводится к выделению групп пикселей, получивших на предыдущем этапе промежуточное значение, и отнесению их к границе (если они соединены с одной из установленных границ) или их подавлению (в противном случае). Пиксель добавляется к группе, если он соприкасается с ней по одному из 8-ми направлений.

Реализация поиска градиента функции

Оператор Собеля выполняет измерение градиента 2D-изображения. Используется пара 3×3 масок свертки: оценки градиента в направлении x (столбцы) и оценки градиента в y -направлении (строк). Данные оценки представлены ниже:

-1	0	+1	+1	+2	+1
-2	0	+2	0	0	0
-1	0	+1	-1	-2	-1
G _x			G _y		

Рис. 1. Оценки градиента

Величина, или край силы, градиента затем аппроксимируется по формуле:

$$|G| = |Gx| + |Gy|,$$

то есть $G = \sqrt{Gx^2 + Gy^2}$.

Ниже представлена программная реализация поиска градиента функции с помощью оператора Собеля. В данной программе, в целях уменьшения вычислительных затрат, изображение изначально преобразуется в оттенки серого. Подобный подход используется во многих методах обработки изображения.

```
__global__ void Kernel(float *matrix, float *fmatrix,
int sizeX, int sizeY)
{
int idx=blockIdx.x * blockDim.x + threadIdx.x;
int idy=blockIdx.y * blockDim.y + threadIdx.y;
int id = idx*sizeX+idy;
```

```
double sobel[3][3];
double sobel1[3][3];
```

```
double sum=0;
for (int fi=0; fi<3; fi++)
for(int fj=0; fj<3; fj++)

sum+=matrix[(idy+fj)*sizeX+idx+fi]*sobel[
fi][fj];
```

```
double sum1 =0;
for (int fi=0; fi<3; fi++)
for(int fj=0; fj<3; fj++)

sum1+=matrix[(idy+fj)*sizeX+idx+fi]*sobel1[fi][fj];
fmatrix[id] = sqrt(sum*sum+sum1*sum1);
}
```

В данной программе: **__global__ void Kernel** – функция, описывающая алгоритм фильтра Собеля; **sum+=matrix[(idy+fj)*sizeX+idx+fi]*sobel[fi][fj];** и **sum1+=matrix[(idy+fj)*sizeX+idx+fi]*sobel1[fi][fj];** – вычисление составляющих градиента Собеля по x и y с помощью матриц.

Результат выполнения



Рис. 2. Исходное изображение

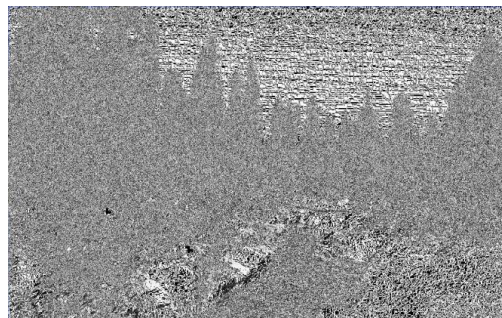


Рис. 3. Обработанное изображение

Результат работы оператора Собеля показывает, насколько «резко» или «плавно» меняется яркость изображения в каждой точке.

Заключение

Поиск градиента позволяет получить вероятность нахождения точки на грани изображения, а также ориентацию границы. Недостатком данного метода заключается в том, что аппроксимация градиента достаточно грубая, что особенно проявляется на высокочастотных колебания изображения.

Литература

1. Боресков А.В. , Харламов А.А. Основы работы с технологией CUDA. – М.: ДМК Пресс, 2010. – 232 с.: ил. ISBN 978-5-94074-578-5.
2. ИНТУИТ. Детектор ребер Канни [Электронный ресурс]. – Режим доступа: <http://www.intuit.ru/studies/courses/10621/1105/lecture/12025?page=7>, свободный.
3. Дрексельская лаборатория автоматизированных систем. «Алгоритм выделения контуров CANNY» [Электронный ресурс]. – Режим доступа: <http://masters.donntu.edu.ua/2010/fknt/chudovskaja/library/article4.htm>, свободный
4. Потапов А.А. Новейшие методы обработки изображений /А.А. Потапов, А.А. Пахомов, С.А. Никитин, Ю.В. Гуляев. – М.: Физматлит, 2008. – 496 с.
5. Лукьяница А.А., Шишкин А.Г. Цифровая обработка видеоизображений. – М.: «Айс-Эс-Эс Пресс», 2009. -518 с. ISBN 978-5-9901-899-1-1