З появою таких API-інтерфейсів, як Open GL або NVIDIA Compute Unified Device Architecture (CUDA), необроблені дані більше не потрібно перетворювати, а програми можна безпосередньо передавати в графічний процесор, що спрощує їх використання. Серед властивостей таких програм для обробки на GPU можна назвати:

1) Мінімальна кількість складних для обробки операцій: ділення, піднесення в степінь і т.д;

2) Відсутність в алгоритмах множинного розгалуження;

3) Невеликий обсяг даних, переданих до відеокарти і від неї в оперативну пам'ять CPU.

Технологія підтримується відеокартами на чіпах NVIDIA, починаючи з 8 серії і новіше, включаючи, Quadro і Tesla.

Варіанти прискорювачів Tesla (рис.2) для робочих станцій відрізняються від відеокарт великою кількістю пам'яті і наявністю тільки одного відеовиходу.

Рис. 2. Прискорювач GPGPU Tesla C2070 з підтримкою CUDA

Така технологія використовується для проектів, що мають розподілені обчислення. Такі проектів використовуються від пошуку радіосигналів позаземного розуму (SETI @ home) до дослідження причин виникнень хвороб людини (Folding @ home) [7].

Отже, наявність відеокарт надає можливість розпаралелювати задачі, а CUDA NVIDIA спрощує використання необроблених даних, передає програми у графічний процесор. Технологією CUDA може користуватися програміст, що знає мову Сі.

Комп'ютерні обчислення загального призначення працюють на основі передачі великих обсягів окремих даних з ЦП в ГП через графічний конвеєр.

Розглянемо основні вимоги для ефективного виконання GPU.

Аналіз технологій обробки процесів, заданих на відеокартах, архітектурних рішень CPU і GPU, а також проведені експериментальні дослідження показали, що в першому чергу пропонуються ті завдання, які добре розпаралелюються на множинні потоки.

Велике прискорення можна отримати, якщо одні і ті ж інструкції застосовуються до великих масивів даних. Наступна вимога - відсутність взаємодії між потоками, що обробляються, або «слабка» взаємодія.

Ще одна вимога: мінімальна кількість складних для обробки операцій: ділення, піднесення у від’ємну степінь і т.д . Важливими вимогами є відсутність в алгоритмах множинних перевірок; невеликі об'єми даних, які передаються до відеокарти і від неї в оперативну пам'ять процесора. Вказані вимоги можна частково обійти за рахунок аналізу алгоритму рішення задач та використання додаткових засобів, які знижують вплив вказаних вимог.

Використання графічних відеокарт для неграфічних розрахунків дозволяє для багатьох застосувань суттєво підвищити продуктивність обчислювального процесу.

Література.

1. Романюк О. Н. Довгалюк Р. Ю., Олійник С. В. Класифікація графічних відеоадаптерів.Наукові праці Донецького національного технічного університету. Сер. : Інформатика, кібернетика та обчислювальна техніка. - 2011. - Вип. 14. - С. 211-215.

2. Полетаев С. А. Параллельные вычисления на графических процесорах. – Режим доступу до ресурсу: https://www.iis.nsk.su/files/articles/sbor_kas_16_poletaev.pdf.
3. Cuda Zone – Режим доступу до ресурсу: https://developer.nvidia.com/cuda-zone
4. CPU vs GPU –Режим доступу до ресурсу: https://www.omnisci.com/technical-glossary/cpu-vs-gpu
5. Вычисления на GPU: мифы и реальность – Режим доступу до ресурсу: https://compress.ru/article.aspx?id=23724
6. Nvidia CUDA неграфические вычисления на графических процесорах – Режим доступу до ресурсу: https://www.ixbt.com/video3/cuda-1.shtml
7. Неграфические вычисления на видеокарте (NVIDIA CUDA и AMD Stream) –Режим доступу до ресурсу: https://poisk-podbor.ru/prices/videokarty/articles/negraficheskie-vychisleniya-na-videokarte-nvidia-cuda-i-amd-stream.
8. Буза М.К. Параллельные вычисления на графических процесорах. –Режим доступу до ресурсу: http://dspace.nbuv.gov.ua/bitstream/handle/123456789/59424/08-Buza.pdf?sequence=1.

УДК 004.92

Романюк О.Н.[1], *д.т.н., професор кафедри программного обеспечения*
Вяткин С.И.[2], *к.т.н.,с.н.с.*
Станиславенко Є.Г.[1], *студент 1 курса специальности «програмного обеспечение»*

3D-МОДЕЛИРОВАНИЕ НА ОСНОВЕ ОДНОГО РАКУРСА ИЗОБРАЖЕНИЯ

[1]Вінницький національний технічний університет, Україна
[2]Институт автоматики и электрометрии СО РАН, Россия

Введение. Генерация моделей трехмерного мира из множества образов является основой компьютерного зрения. Интересным предельным случаем является проблема реконструкции по одному ракурсу изображения. Это крайне некорректная задача, где стерео и соответствие точек не могут быть применены. Тем не менее, это важная проблема: во многих приложениях имеется только единственное изображение сцены, а необходимо интерактивно извлечь твердое тело 3D модели соответствующих объектов для приложений виртуальной и дополненной реальности. Существуют подходы, направленные на реконструкцию полей высот [1], но они не подходят для получения замкнутых 3D-поверхностей. В работе [2] предложен метод способный расширить подход для поверхностей с одним или двумя отверстиями, однако он не является обобщением на объекты произвольной топологии.

Данная работа посвящена проблеме реконструкции из одного ракурса изображения. В предлагаемом методе можно вычислить непротиворечивые по силуэту взвешенные минимальные поверхности для пользовательского объема, используя методы выпуклой релаксации.

Описание метода. Предположим, что дан силуэт объекта на изображении, полученный с помощью интерактивного инструмента сегментации. Цель состоит в том, чтобы получить гладкую 3D-модель, которая соответствует силуэту. Как выбрать правильный вариант 3D модели среди бесконечно многих, которые соответствуют силуэту? Для этого необходимо иметь дополнительную информацию, в то же время эта информация должна быть минимальной. Достаточно просто указать объем объекта и вычислить минимальную поверхность заданного объема для возникновения семейства правдоподобных 3D-моделей. Пусть дана плоскость изображения P, которая содержит входное изображение и лежит в $\mathfrak{R}^3$.

В качестве части изображения также имеем силуэт объекта $s \subset P$. Необходимо вычислить реконструкции как минимальные взвешенные поверхности $S \subset \Re^3$, которые имеют определенный целевой объем V и соответствуют силуэту объекта s:

$$\min \int_S f_{sw}(e) ds,\ \pi(S) = e,\ V_{es}(S) = V \tag{1}$$

Где $\pi : \Re^3 \to P$ - ортогональная проекция на плоскость изображения P, $f_{sw} : \Re^3 \to \Re^+$ - функция взвешивания гладкости, $V_{es}(S)$ - объем, окруженный поверхностью S и $e \in S$. $e \in S$ - это элемент поверхности.

Минимальная взвешенная площадь поверхности задается путем минимизации общего количества, подходящего множества Set допустимых функций f_f:

$$\min_{f_f \in Set} \int_S f_{sw}(x) \left| \nabla f_f(x) \right| d^3 x \tag{2}$$

Где ∇f_f - производная в дистрибутивном смысле. Тогда все поверхностные функции, в соответствии с силуэтом должны быть в комплекте

$$Set_s = \left\{ f_f \in bound_{\text{var}}\left(\Re^3; \{0,1\}\right) \middle| \ f_f(x) = \begin{cases} 0, \ \pi(x) \notin s \\ 1, \ x \in s \end{cases} \right\} \tag{3}$$

Решая уравнения (3), относительно множества последовательных функций силуэта результатом будет сам силуэт. Для решения уравнения (2), предлагается использовать ограничение на размер объема, заключенного в минимальную поверхность. При ограничении допустимого множества Set_s реконструируемая поверхность должна иметь определенный целевой объем V

$$\min_{f_f \in Set_s \cap Set_V} E(f_f) \tag{4}$$

где

$$E(f_f) = \int f_{sw}(x) \left| \nabla f_f(x) \right| d^3 x \tag{5}$$

и

$$Set_V = \left\{ f_f \in bound_{\text{var}}\left(\Re^3; \{0,1\}\right) \middle| \int f_f(x) d^3 x = V \right\} \tag{6}$$

где Set_V обозначает все реконструкции с ограниченной вариацией, которые имеют специфический объем V.

Функция инфляции объема δ_{vi} позволяет сделать некоторое предположение о форме объекта. Для любой точки $p \in P$ пусть

$$\left\{ d_f\left(p, \partial S\right) \right\} = \min_{s \in \partial S} \left\| p - s \right\| \tag{7}$$

Определяет расстояние до силуэта контура $\partial S \subset P$. Затем:

$$\delta_{vi}(x) = \begin{cases} +1, \ otherwise \\ -1 \quad if \ d_f(x, P) \le h(\pi(x)) \end{cases} \tag{8}$$

Карта высот $h : P \to \Re$ определяется с помощью функции расстояния

$$h(p) = \min \left\{ v_{co}, v_{os} + v_f d_f\left(p, \partial S\right)^n \right\} \tag{9}$$

Параметры $\left\{ v_{co}, v_{os}, v_f, n \right\}$ модифицируют форму h.

Заключение. В предлагаемом методе используется неявное представление поверхности, заданное индикаторной функцией. Взвешенная минимальная поверхностная задача представляет собой выпуклый функционал и релаксацию двоичной функции, что приводит к общей выпуклой задаче. Ограничение объема сводится к выпуклому ограничению, которое легко интегрируется в процессе реконструкции. Индикаторная функция релаксации