

НАХОЖДЕНИЕ ЗАДАННОГО ОБЪЕКТА НА ВИДЕО С ПОМОЩЬЮ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ

Е.Н Чеботарева, С.В. Аксёнов
Томский политехнический университет
enc1@tpu.ru

Введение

В настоящее время, в связи с развитием информационных технологий и систем автоматизации, растёт потребность в обработке информации, в частности такой как видео и изображения. Обработка видео и изображений развита достаточно, чтобы найти широкое применение своих достижений в различных сферах.

Достаточно назвать такие области применения, как интеллектуальные робототехнические системы, исследование природных ресурсов из космоса, биомедицинские исследования, криминалистика, сжатие и передача изображений по узкополосным каналам связи, автономная навигация, оптическая локация, управление движением воздушных и наземных транспортных средств, обнаружение заданных объектов и символов в информационно-поисковых системах, автоматическое обнаружение объектов в охранных системах, системах самонаведения и т.д. [1] Все эти области связаны с компьютерным зрением.

Компьютерное зрение — теория и технология создания машин, которые могут производить обнаружение, слежение и классификацию объектов.

Как научная дисциплина, компьютерное зрение относится к теории и технологии создания искусственных систем, которые получают информацию из изображений. Видеоданные могут быть представлены множеством форм, таких как видеопоследовательность, изображения с различных камер или трехмерными данными, например с устройства Kinect или медицинского сканера. [2]

Задача поиска имеет большое значение для вышеупомянутых сфер человеческой деятельности. Поиск заданного объекта на видео в сущности сводится к поиску на статическом изображении, так как видеофайл представляет собой последовательность кадров (изображений), лишь с тем отличием, что он осуществляется многократно для некоторого количества кадров.

Существует несколько подходов к решению проблемы поиска объекта в видеофайлах или на растровых изображениях. Алгоритмы поиска по цвету, по текстуре или по форме. Поиск по текстуре ложно формально описать, а поиск по форме сложно полностью автоматизировать для широкого класса задач. Самым оптимальным вариантом является поиск по цвету, так как он не зависит от размеров или ориентации изображения. Для его осуществления изображение делится на регионы со сходными цветовыми характеристиками.

Для упрощения задачи необходимо обработать изображение, содержащее искомый объект — шаблон для поиска. Для этих целей применяется алгоритм Кохонена [3], с помощью которого происходит кластеризация - группировка, разбиение множества объектов на непересекающиеся подмножества, кластеры, состоящие из схожих объектов. В данном случае кластерами являются цвета. В изображении-шаблоне выделяются основные цвета, по которым и будет проводиться поиск. Далее изображение перекрашивается в эти главные цвета и оно используется для дальнейшего анализа. Это позволит укорить обнаружение объекта.

При таком подходе удобным решением является использование цветовой модели RGB. Пиксели шаблона и кадра видео представляются в виде последовательностей RGB.

Для получения кадра видео, удобного представления кадра и шаблона в RGB и получения их размеров можно использовать OpenCV (Open Source Computer Vision Library) - библиотеку компьютерного зрения с открытым исходным кодом. Также данная библиотека позволяет осуществить смену кадров для их анализа.

Применение параллельных вычислений для поиска объекта по шаблону в видеофайлах

Данная задача связана с обработкой графической информации, поэтому логичным является задействовать для ее решения ресурсы видеокарты, точнее графического процессора — GPU (graphics processing unit). Это осуществляется с помощью технологии CUDA (Compute Unified Device Architecture).



Рис.1. Расчет выделяемой памяти

Для обмена данными при работе центрального и графического процессоров необходимо явно выделять память RAM -> GPU и GPU -> RAM. Расчет необходимой выделяемой памяти происходит так: изображение — это массив RGB, поэтому количество пикселей необходимо умножить на 3 байта.

Функция ядра выполняется графическим процессором и для того, чтобы выполнить поставлен-

ную задачу быстрее, она может использовать параллельные вычисления. Технология CUDA различает три понятия: сетка, блок и поток (рис.2).

Считается, что в одной сетке могут содержаться блоки по двум измерениям от 1 до 65536 по каждому, а каждый из блоков может состоять примерно около $512 * 512 * 64$ нитей/потоков (по трем измерениям). [4] Функция ядра выполняется каждой нитью (потоком) каждого блока сетки.

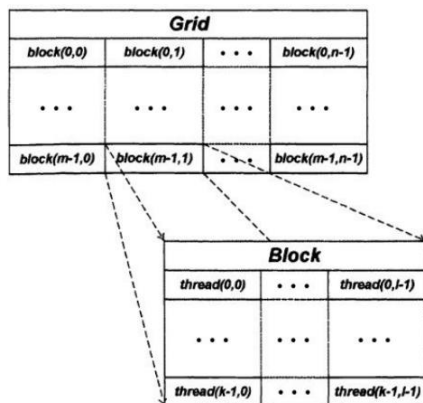


Рис.2. Иерархия нитей CUDA

Для того, чтобы определить какой именно блок, или какая именно нить исполняет этот код, используются встроенные переменные:

1. `threadIdx.x` – номер потока по измерению x .
2. `blockIdx.x` – номер блока по измерению x .
3. `threadDim.x` – число нитей по измерению x в блоке.
4. `blockDim.x` – число блоков по измерению x в блоке. [4]

Функция ядра осуществляет следующее:

1. шаблон каждым блоком «совмещается» с исходным изображением в различных местах;
2. все нити каждого из блока сравнивают цвет каждого пикселя шаблона с цветом каждого пикселя части исходного изображения, где для данного блока в данный момент времени расположен шаблон;
3. если расстояние между сравниваемыми пикселями в цветовой модели RGB (кубе) меньше 50 ($r = [0; 255]$, $g = [0; 255]$, $b = [0; 255]$), то поток прибавляет единицу к своему значению в разделяемом между потоками на блок массиве `summa`, а иначе – не прибавляет; после того, как массив `shared` заполнился потоками, они синхронизируются;
4. и далее происходит суммирование всех ячеек массива `shared` параллельным образом путем редуцирования;
5. для данного положения шаблона на изображении каждый блок (массив нитей) получает число близких пикселей; при этом, если число близких пикселей по отношению к общему числу

пикселей равно 50% или больше, то считается, что объект распознан;

6. рисуется рамка, выделяющая результат поиска, для этого задействуется множество нитей.

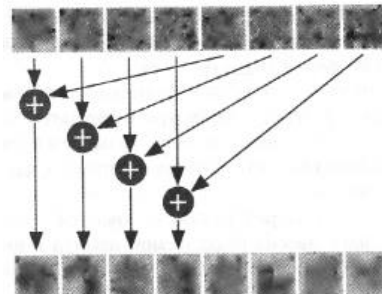


Рис. 3. Один шаг суммирования путем редукиции

Стоит отметить, что за счет параллельного суммирования массива `shared` путем редуцирования (как частный случай уменьшения массива) суммирование происходит гораздо быстрее.

Заключение

Для поиска объекта по шаблону в видеофайле выполняются следующие действия: кластеризация искомого изображения (алгоритм Кохонена), получение кадра видео (OpenCV), обработка кадра и изображения и их сопоставление (CUDA), получение результатов поиска. Параллельные вычисления, реализованные с помощью технологии CUDA позволяют достичь высокого быстродействия алгоритма при обработке больших объемов графической информации. Полученное решение может быть применено в различных сферах деятельности благодаря высокой производительности.

Литература

1. Б. А. Алпатов, П. В. Бабаян, О. Е. Балашов, А. И. Степашкин – Методы автоматического обнаружения и сопровождения объектов. Обработка изображений и управление - М.: Радиотехника, 2008 г. – 176 с.
2. Википедия - Компьютерное зрение [Электронный ресурс]. Режим доступа: http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%BE%D0%B5_%D0%B7%D1%80%D0%B5%D0%BD%D0%B8%D0%B5, свободный.
3. Экостат [Электронный ресурс]. Режим доступа: <http://www.ievbras.ru/ecostat/Kiril/Library/Book1/Content396/Content396.htm, свободный>
4. Сандере Дж., Кэндрот Э. - Технология CUDA в примерах: введение в программирование графических процессоров - М.: ДМК Пресс, 2011 г. – 232 с.