Самый простой подход для реализации свертки на CUDA – это загрузка изображения в массив в совместно используемой памяти (Shared memory), выполнение поточечного перемножения блоков, на которые было разбито изображение, затем, изменения вносятся в изображение в памяти видеокарты. Каждая нить в блоке обрабатывает один блок в изображении. Каждая нить генерирует один выходной пиксель. Это показано на рисунке 3.

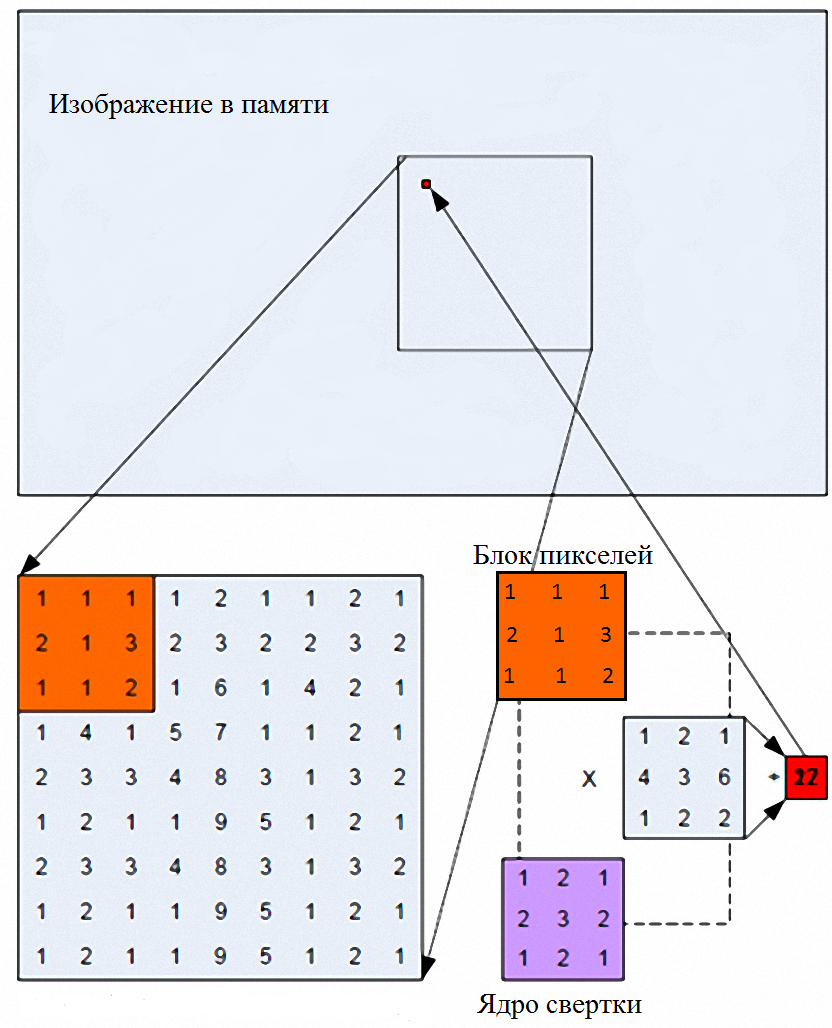


рис. 3

Описание рисунка 3: алгоритм свертки. Блок пикселей изображения загружается в массив в совместно используемой памяти (shared memory). Для обработки и расчета выходного пикселя (красный) область входного изображения (orange) умножается поэлементно на ядро фильтра (пурпурный) и результаты суммируются. Получившийся пиксель записывается обратно в изображение.

Алгоритм сам по себе несколько сложнее. Для любого возможного размера ядра фильтра, пиксели на краю массива разделяемой (shared) памяти будут зависеть от пикселей в неразделяемой памяти. Вокруг блока изображения в пределах блока нитей располагается граница из пикселей, по ширине радиуса ядра, необходимого для фильтрации блока изображения. Таким образом, каждый блок нитей должен загрузить в разделяемую память (shared memory) пиксели, которые должны быть отфильтрованы и пиксели границы. Это демонстрируется на рисунке 4. Граница одного блока перекрывается с соседними блоками. Границы блоков на краях изображения выходят за его пределы – эти пиксели могут быть прикреплены к цвету пикселей на краю изображения, либо установлены в ноль.