

Задачи для студентов CBG Hardware

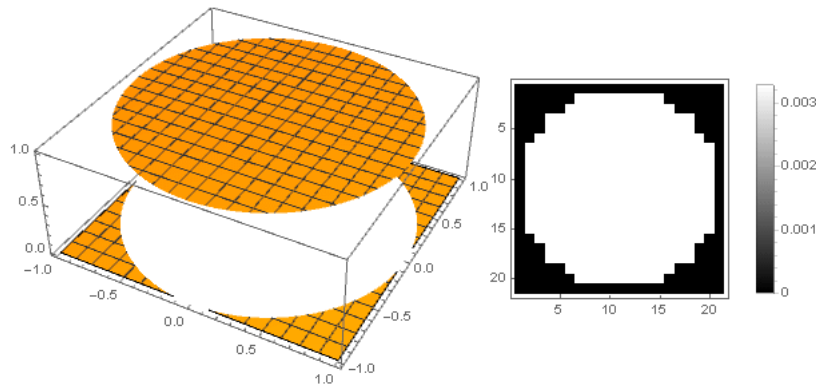
Радькин Кирилл Алексеевич, группа Б01-005

1. По теореме Котельникова, частота дискретизации должна быть в 2 раза (или больше) превосходить максимальную частоту передаваемого сигнала, соответственно необходимая частота дискретизации: 30 кГц. Количество уровней квантования определяется динамическим диапазоном. На 6 дБ глубина квантования 1 бит, тогда в нашем случае будет 10 бит (т.к. 60 дБ). Глубина квантования определяет, каким количеством бит будет закодирован один уровень квантования. Т.к. у нас получается 10 бит, тогда уровней может быть $2^{10} = 1024$. Дополнительная обработка сигнала может быть необходима, например, чтобы отсеять шумы, мы можем применить фильтр верхних/нижних частот.
2. Сложность подсчета одного элемента итоговой матрицы L (нам нужно перемножить L пар чисел и сложить их). Всего в итоговой матрице $M \times N$ элементов, тогда получаем сложность $M \times N \times L$. Теоретически мы можем считать каждое произведение на отдельном процессоре (их как раз L), после этого нам нужно как-то сложить эти произведения. Один из вариантов, загрузить все произведения в один процессор и сложить в нем, но тогда одинаковое количество операций возможно лишь в случае, если произведение $M \times N$ делится на L . В другом случае, мы можем загружать нашу сумму в память, т.е. каждый процессор будет считывать уже подсчитанное значение для элемента матрицы, прибавлять к нему то, что он подсчитал и загружать обратно. В таком случае кол-во операций будет одинаковым, однако теряется весь смысл распараллеливания (т.к. каждому процессору придется ждать, пока другие процессоры работают с необходимым ему элементом матрицы).
3. Один из первых моментов, которые я отметил для себя в этом задании, это то, что после свертки (4 раза) размер тензора не изменился, хотя свертка как раз таки уменьшает размер тензора. Соответственно, приходим к выводу, что присутствует padding (искусственное увеличение размера входного тензора). Путем нехитрых вычислений, учитывая, что размер тензора после прохождения одного слоя не изменяется (т.к. padding применяется перед прохождением каждого слоя), учитывая, что размер ядра равен 3, получаем, что $\text{padding} = 2$. Т.к. у нас ядро свертки 3×3 , получаем 9 обучаемых параметров на каждом слое (плюс возможно линейное отклонение, тогда 10). Тогда, всего 4 слоя + на каждый канал свое ядро, следовательно имеем 72 параметра (80). Т.к. отсутствуют функции активации, любое количество подряд идущих линейных преобразований (а свертка является линейным преобразованием), можно заменить одним линейным преобразованием. Можно провести аналогию с перемножением матриц, когда любую последовательность $A_1 \times A_2 \times A_3 \dots \times A_n \times X$ (где X - некий вектор, а A_i - матрицы) можно заменить одним произведением $A \times X$. Соответственно, в нашей задаче, мы можем заменить 4 сверточных слоя на 1, не потеряв при этом в качестве (и при этом, очевидно, ускорив обучение).
4. Данный компилятор может применять следующие оптимизации. Первая и одна из самых очевидных: свертывание констант, т.е. условная строчка $x + 10 \cdot 2 + 3$ может

быть свернута в $x + 23$, тогда байт код программы очевидно измениться, т.к. процессору не нужно больше вычислять $10 \cdot 2 + 3$, исходный код программы при этом не изменяется. Также такой компилятор может использовать SSE инструкции, т.е. производить сразу несколько вычислений одинаковых вычислений за один такт процессора, но, насколько мне известно, компиляторы редко отлавливают такие места кода, в которых можно это применить. Исходный код при этом также не изменяется. Кроме того, программист может явно указать в своем коде, что хочет использовать SSE инструкции (например с помощью ассемблерной вставки), тогда компилятор будет их использовать. В таком случае, исходный код программы изменяется.

5. Т.к. кожа на фотографии стала гладкой, очевидно был применен какой-то алгоритм размытия. Возможно, автор алгоритма задал слишком резкую функцию затухания (как на скриншоте ниже).

$$\begin{cases} 1, & x < 1 \\ 0, & x \geq 1 \end{cases}$$



Источник картинки <https://habr.com/ru/post/432622/>

При использовании такого (или похожего фильтра) будет получаться данный по условию эффект размытия (с потерей оттенков цветов кожи, теней и т.д.) Для решения этой проблемы можно, например добавить зависимость затухания от координаты. Т.е. некоторые участки кожи будут казаться темнее или светлее. Один из таких алгоритмов: размытие по Гауссу.