Вычисление свертки при помощи теоретико-числовых преобразований.

Денис Морозов

27 марта 2012 г.

Полученный результат:

Реализована целочисленная операция двумерной свертки для размеров $2^m \times 2^n$, где $0 \leqslant m, n \leqslant 10$.

1 Преобразования Фурье над конечными полями.

Рассмотрим мотивацию вычисления свертки над конечными полями вместо поля $\mathbb{C}$ комплексных чисел.

При применение свертки к фильтрации изображений входные данные являются целыми числами, и применение целочисленной арифметики для работы с ними является естественным. При вычислении свертки над конечным полем, в отличии от поля $\mathbb{C}$, отсутствует накопление погрешности вычислений.

Результат свертки последовательности

$$0\ 0\ 0\ 0\ 0\ 126\ 126\ 126\ 126\ 126\ 0\ 0\ 0\ 0\ 0\ 126\ 126\ 126\ 126\ 126\ 0\ ...$$

с ядром $-1\ 2\ 1$ с применением преобразования Фурье
над полем $\mathbb{C}$ комплексных чисел:

$$0.00000\ 0.00000\ -0.00001\ -0.00002\ 0.00002\ 126.00000\ -125.99998\ 0.00001$$

$$0.00002\ -0.00000\ -125.99998\ 126.00002\ -0.00002\ -0.00000\ -0.00001\ 125.99999...$$

над конечным полем $GF(p)$:

$$0\ 0\ 0\ 0\ 0\ 126\ -126\ 0\ 0\ 0\ 126\ -126\ 0\ 0\ 0\ 126\ ...$$

Скорость же работы с целыми числами типа int для большинства процессоров значительно превышает скорость роботы с данными типа float и double, с помощью которых реализуется арифметика поля $\mathbb{C}$. Кроме того, как будет показано далее, при удачном выборе генератора преобразования Фурье над конечным полем умножение может быть заменено двоичными сдвигами, что тоже значительно уменьшает время выполнения преобразования. Так же вычисление обратного преобразования

Фурье содержит сложную операцию деления, которую в конечном поле можно заменить умножением на целый обратный элемент поля.

К сложности реализации преобразования Фурье над конечным полем следует отнести необходимость использования операции взятия по модулю размера поля. Скорость работы алгоритма непосредственно зависит от простоты реализации данной операции и от частоты ее использования.

1.1 Преобразования Ферма над конечным полем.

Рассмотрим преобразования Фурье над конечным полем Галуа $GF(2^{16}+1)$. Возьмем элемент w порядка 1024, например $w = 18729$. Найдем обратный к нему $w^{-1} = 27119$. w и w^{-1} являются генераторами соответственно для прямого и обратного преобразования Фурье длины 1024 над полем $GF(2^{16}+1)$. Преобразование длины 2^n задается генератором $w^{2^{10-n}}$:

$$w^1 = 18729,\ w^{-1} = 27119,\ length = 1024$$

$$w^2 = 21417,\ w^{-2} = -16053,\ length = 512$$

$$w^4 = -5574,\ w^{-4} = 7325,\ length = 256$$

$$w^8 = 4938,\ w^{-8} = -19178,\ length = 128$$

$$w^{16} = 4080,\ w^{-16} = 2040,\ length = 64$$

$$w^{32} = 2,\ w^{-32} = -32768,\ length = 32$$

Генератор преобразования длины 32 равен 2, следовательно умножения в преобразовании Фурье можно заменить сдвигами.

Вычислим свертку длины 1024 методом Кули-Тьюки по схеме 32×32.

По классической схеме преобразование длины 32 можно также разбить на преобразования длины k_1 и k_2, где $k_1 * k_2 = 32$. Однако при таком подходе возрастает количество операций взятия по модулю поля $GF(2^{16}+1)$, кроме того добавляются операции, используемые для трансформации матрицы в алгоритме Кули-Тьюки, включающие операции измерения индексов подстановки и присваивания, что существенно увеличивает время выполнения алгоритма. Дальнейшее снижение сложности может быть получено за счет использования для вычисления преобразований Фурье длины 32 вместо алгоритма Кули-Тьюки другого алгоритма.

Для использования нового алгоритма предварительно обработаем матрицу преобразования. Запись матрицы преобразования длины 32 в поле $GF(2^{16}+1)$ в виде неотрицательных остатков по модулю $2^{16}+1$ имеет несколько недостатков.

Во первых неотрицательные остатки степеней двойки, больших $2^{16}+1$ степенями двойки не являются, и мы лишаемся преимущества замены умножения сдвигами. Во вторых двоичное представление наибольшего положительного остатка 2^{16} занимает 17 битов, поэтому при умножении на матрицу преобразования возможно переполнение 32-битных регистров ($2^{16} * 2^{16} = 2^{32}$ - результат занимает 33 бита) и приходится использовать длинную арифметику, что существенно увеличивает время выполнения алгоритма.

Для того, что бы избавиться от вышеперечисленных недостатков отнимем от всех элементов матрицы преобразования, больших 2^{15}, $2^{16}+1$ столько раз, сколько необходимо, что бы данный элемент попал в диапазон $[-2^{15} \dots 2^{15}]$.

Поскольку множество остатков по модулю $2^{16}+1$ с операциями деления и умножения образует кольцо, то данная операция не повлияет на результат вычисления свертки, при условии, что входные и выходные данные также будут находиться в диапазоне $[-2^{15} \dots 2^{15}]$.

После применения данной операции для вычисления свертки достаточно 32 - битной арифметики. Действительно, произведение двух чисел из диапазона $[-2^{15} \dots 2^{15}]$ в двоичном представлении содержит 31 бит плюс один бит на знак($-2^{15} \cdot 2^{15} = -2^{30}$ - результат занимает 32 бита).

Кроме того все элементы матрицы преобразования становятся степенями двойки, быть может умноженные на минус. Действительно, пусть элемент матрицы имеет вид

$$2^{16+t} - k' * (2^{16}+1) \geqslant 0$$

Преобразуем его к виду

$$2^{15} \geqslant 2^{16+t} - k'' * (2^{16}+1) \geqslant -2^{15}$$

Положим $k'' = 2^t$. Дальше

$$2^{16+t} - k'' * (2^{16}+1) = 2^{16+t} - 2^t * (2^{16}+1) = -2^t$$

Поскольку $2^{32} \equiv 1 \ mod(\ 2^{16}+1)$, то можно считать, что $t \leqslant 15$, следовательно $-2^t \geqslant -2^{15}$ ч.т.д.

Прямое вычисление преобразования Фурье при помощи матрицы, обработанной вышеприведенным методом содержит 992 сложения, 961 нетривиальный сдвиг вместо умножений, 64 операций быстрого взятия по модулю, реализованного, как один сдвиг, одно побитовое пересечение и одно вычитание, и 32 операций взятия по модулю, реализованного, как три сдвига, одно побитовое пересечение, три вычитания и одно сложение. То есть для вычисления прямого преобразования необходимо 1024 = 992 + 32 сложения, 160 вычитаний, 1121 = 961 + 160 сдвиг и 96 побитовых пересечений.

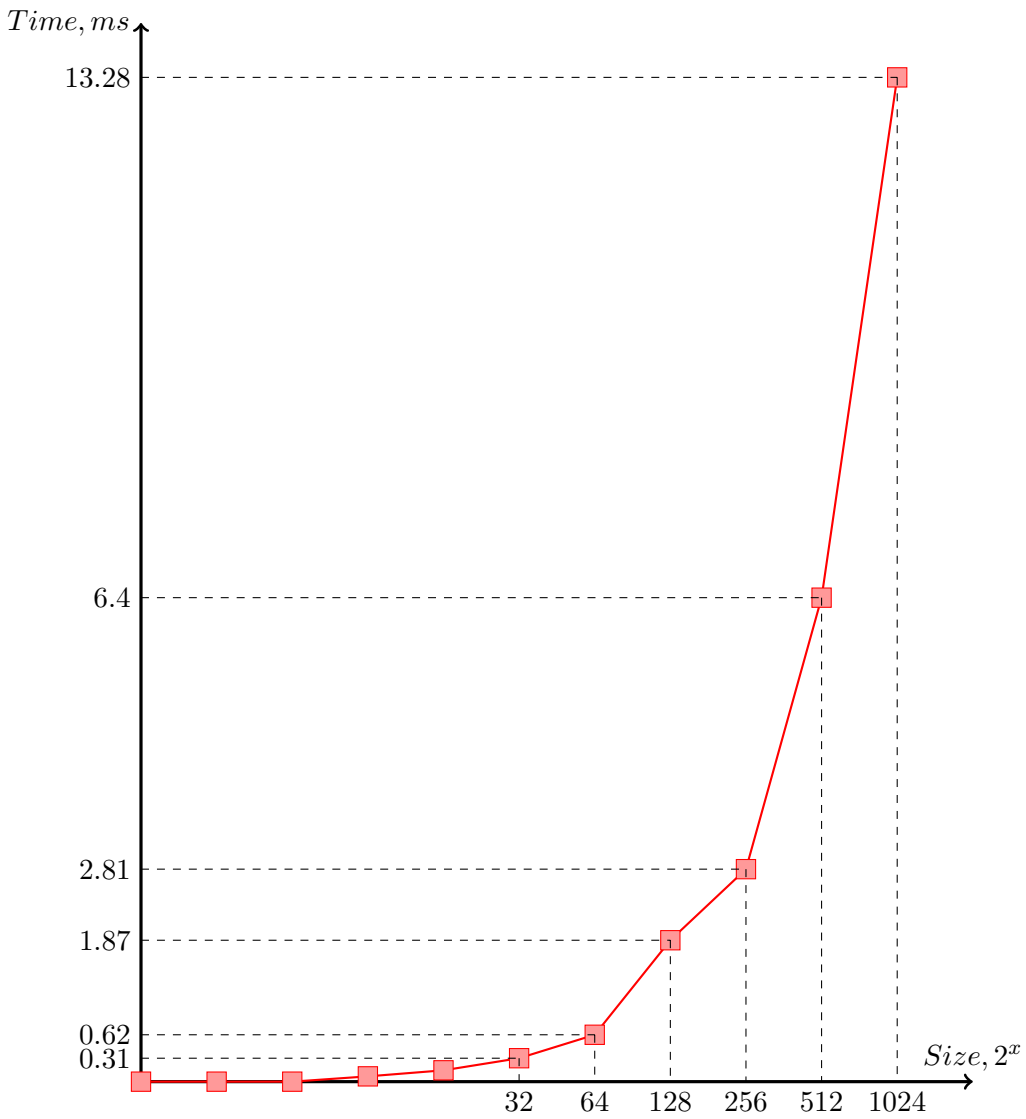
Заметим, что после обработки данным методом матрица приобретает определенную симметрию по строкам и столбцам, которую можно использовать для уменьшения количества операций при вычислении преобразования длины 32.

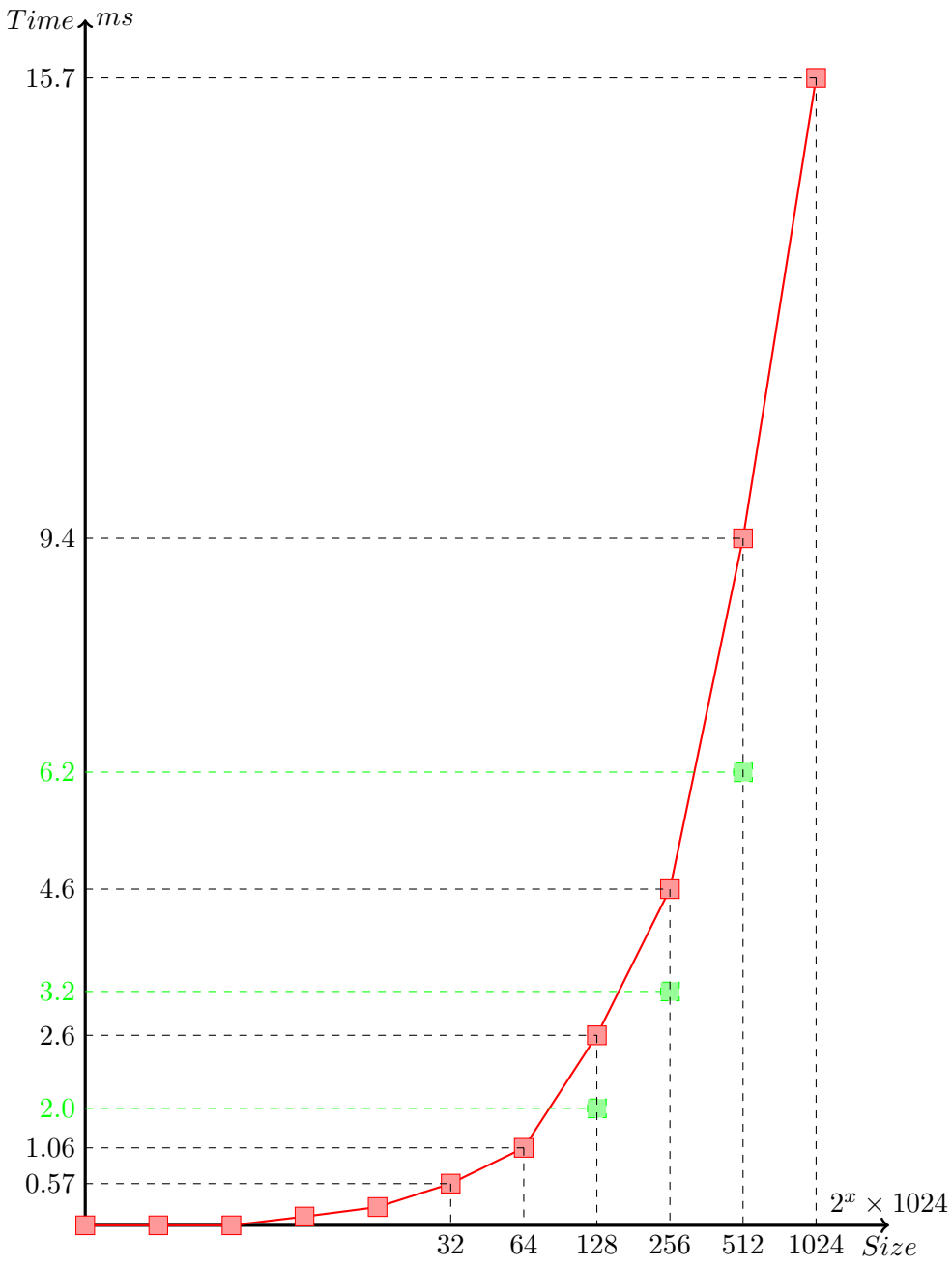
Алгоритм, использующий симметрию для вычисления прямого преобразования длины 32 использует 190 = 158 + 32 сложений, 266 = 106 + 160 вычитаний, 315 =

155 + 160 сдвигов и 96 побитовых пересечений, что сравнимо с количеством операций, используемых при итеративном использовании алгоритма Кули-Тьюки. (По сравнению с алгоритмом Кули-Тьюки 2×16 данный алгоритм на 30% быстрее. Данные относительной скорости алгоритмов для процессора Intel 2.9Ghz: Кули-Тьюки - 47, используемый алгоритм - 31).

Основной алгоритм комплексного БПФ по основанию два		Полностью оптимизированный алгоритм комплексного БПФ по основанию два		алгоритм Рейдера-Бреннера комплексного БПФ по основанию два		Алгоритм БПФ, использующий симметрию матрицы преобразования		
Число вещественных умножений	Число вещественных сложений	Число вещественных умножений	Число вещественных сложений	Число вещественных умножений	Число вещественных сложений	Число сдвигов	Число целых сложений	Число побитовых и (&)
320	480	88	408	68	512	315	456	96

Однако при оценке количества операций, используемых при вычисления БПФ итеративным методом Кули-Тьюки необходимо учитывать так же операции, используемые для трансформации матрицы, включающие операции измерения индексов и присваивания, поскольку не известен общий алгоритм вычисления трансформации матрицы по месту в данном методе. Заметим, что в приведенной таблице данные операции не учитываются, а в алгоритме, использующем симметрию, эти операции отсутствуют.


Time, ms
13.28
6.4
2.81
1.87
0.62
0.31
32
64
128
256
512
1024
Size, 2^x



На графике видно, что при увеличение размера входных данных в два раза время вычисления свертки возрастает приблизительно в два раза, что говорит о том, что сложность алгоритма близка к $N \log N$.

1.2 Преобразования Мерсенна над конечным полем.

Преобразования Фурье над следующими полями удобны простой арифметикой взятия по модулю поля:

$GF(2^{13}-1), GF(2^{17}-1), GF(2^{19}-1), GF(2^{31}-1)$.

К сожалению малый порядок двойки в группе по умножению в данных полях не позволяет заменить умножение сдвигами для больших длин преобразования Фурье, как в случае с полями Ферма.

1.3 Преобразования Фурье над кольцом.

Для корректного определения дискретного преобразования Фурье длины n над кольцом необходимо существования элемента w порядка n ($w^n = 1$), при этом n должен быть обратимым элементом мультипликативной подгруппы этого кольца. При данных условиях существует корректно определенное прямое и обратное отображения Фурье над данным кольцом, при помощи которых можно вычислить свертку.

1.4 Технические моменты.

1. Уменьшение количества взятий по модулю.

2. Уменьшение количества операций преобразования Фурье длины 32, использующее структуру матрицы преобразования. Увеличение скорости в 3 раза.

3. Переход от 64-битной арифметике к 32-битной. Увеличение скорости в 5 раз.

Пусть A, B - матрицы размера $m \times n$, R, R^{-1} - матрицы преобразования Фурье над строчками, L, L^{-1} - матрицы преобразования Фурье над столбиками.

Двумерная свертка матриц A, B записывается формулой

$$A \hat{\circ} B = L^{-1}((LAR) \circ (LBR))R^{-1}$$

Поскольку умножение матриц ассоциативно, то

$$A \hat{\circ} B = (L^{-1}((L(AR)) \circ (L(BR)))R^{-1} = ((AR) \hat{\circ}_c (BR))R^{-1} = L^{-1}((LA) \hat{\circ}_r (LB))$$

где $\hat{\circ}_c$ - свертка матриц по столбикам, а $\hat{\circ}_r$ - по строчкам. Следовательно результат двумерной свертки не зависит от выбора генераторов преобразований R, L.

При вычислении преобразований Фурье над конечным полем алгоритмом Кули-Тьюки квадратная форма матрицы предпочтительнее,

поскольку уменьшает количество взятий по модулю размера поля. Кроме того трансформация квадратной матрицы имеет простой алгоритм применения по месту.

Выбор схемы для вычисления FFT длины 64: 8×8

Свертку с большим ядром можно рассматривать, как последовательность сверток с маленьким ядром, предварительно разложив ядро на множители.

$$!1688 \quad 14407 \;\; 14406 \;=\; 2 * 3 * 7^4$$

$!3138 \quad 28813 \;\; 28812 = 2^2 * 3 * 7^4$

$3512 \quad 32749 \;\; 32748 = 2^2 * 3 * 2729$

$32771 \;\; 32770 = 2 * 5 * 29 * 113$

$32779 \;\; 32778 = 2 * 3^3 * 607$

$32783 \;\; 32782 = 2 * 37 * 443$

$3516 \quad 32789 \;\; 32788 = 2^2 * 7 * 1171$

$3517 \quad 32797 \;\; 32798 = 2 * 23^2 * 31$

$!3518 \quad 32801 \;\; 32800 = 2^5 * 5^2 * 41$

$3519 \quad 32803 \;\; 32802 = 2 * 3 * 7 * 11 * 71$

$3520 \quad 32831 \;\; 32830 = 2 * 5 * 7^2 * 67$

$32833 \;\; 32832 = 2^6 * 3^3 * 19$

$32839 \;\; 32838 = 2 * 3 * 13 * 421$

$32843 \;\; 32842 = 2 * 16421$

$3524 \quad 32869 \;\; 32868 = 2^2 * 3^2 * 11 * 83$

$3525 \quad 32887 \;\; 32888 = 2^3 * 4111$

$3526 \quad 32909 \;\; 32908 = 2^2 * 19 * 433$

$3527 \quad 32911 \;\; 32910 = 2 * 3 * 5 * 1097$

$3528 \quad 32917 \;\; 32916 = 2^2 * 3 * 13 * 211$

$!3749 \quad 35153 \;\; 35152 = 2^4 * 13^3$

$!!4144 \quad 39367 \;\; 39366 = 2 * 3^9 \quad (-2)^{2*3^7} = 1$

$!5675 \quad 55903 \;\; 55902 = 2 * 3 * 7 * 11^3 \quad (-2)^{2*3*7*11^3} = 1$

$!!5905 \quad 58321 \;\; 58320 = 2^4 * 3^6 * 5 \quad 2^{2^3 3^5 5} = 1$

$6000 \quad 59359 \;\; 59358 = 2 * 3 * 13 * 761$

$6100 \quad 60497 \;\; 60498 = 2 * 3^2 * 3361$

$6200 \quad 61631 \;\; 61630 = 2 * 5 * 6163$

$!!6276 \quad 62501 \;\; 62500 = 2^2 * 5^6 \quad (-2)^{2^2 * 5^6} =$

$6300 \quad 62773 \;\; 62772 = 2^2 * 3 * 5231$

$6400 \quad 63809 \;\; 63808 = 2^6 * 997$

$!!6426 \quad 64153 \;\; 64152 = 2^3 * 3^6 * 11 \quad 2^{2^2 * 3^5 * 11} = 1$

$$6500 \quad 65063 \;\; 65062 \;=\; 2 \;*\; 32531$$

$$6501 \quad 65071 \;\; 65070 \;=\; 2 \;*\; 3^3 \;*\; 5 \;*\; 241$$

$$6510 \quad 65147$$

$$6520 \quad 65267$$

$$6530 \quad 65381$$

$$6531 \quad 65393 \;\; 65392 \;=\; 2^4 \;*\; 61 \;*\; 67$$

$$6532 \quad 65407 \;\; 65406 \;=\; 2 \;*\; 3 \;*\; 11 \;*\; 991$$

$$6533 \quad 65413 \;\; 65412 \;=\; 2^2 \;*\; 3^2 \;*\; 23 \;*\; 79 \;\; 65413 \; divides \; 34^{23} - 1$$

$$6534 \quad 65419 \;\; 65418 \;=\; 2 \;*\; 3 \;*\; 10903$$

$$6535 \quad 65423 \;\; 65422 \;=\; 2 \;*\; 7 \;*\; 4673$$

$$6536 \quad 65437 \;\; 65436 \;=\; 2^2 \;*\; 3 \;*\; 7 \;*\; 19 \;*\; 41$$

$$6537 \quad 65447 \;\; 65446 \;=\; 2 \;*\; 43 \;*\; 761$$

$$6538 \quad 65449 \;\; 65448 \;=\; 2^3 * 3^4 * 101 \;\; 65449 \;=\; 80^2 + 243^2 \;\; 65449^2 \;=\; 38880^2 + 52649^2 \;\; 65449 \;=\; 2^{16} - 87$$

$$6539 \quad 65479 \;\; 65478 \;=\; 2 \;*\; 3 \;*\; 7 \;*\; 1559$$

$$6540 \quad 65497 \;\; 65498 \;=\; 2 \;*\; 32749$$

$$6541 \quad 65519 \;\; 2 \;*\; 17 \;*\; 41 \;*\; 47$$

$$!6542 \quad 65521 \;\; 65520 \;=\; 2^4 \;*\; 3^2 \;*\; 5 \;*\; 7 \;*\; 13 \quad 65521 = 2^{16} - 15 \;\; 2^{2*3^2*5*13} = 1$$

$$6543 \quad 65537 \;\; 65536 \;=\; 2^{16}$$

$$6545 \quad 65543$$

$$6549 \quad 65579$$

$$!8008 \quad 81901 \;\; 81900 = 2^2 * 3^2 * 5^2 * 7 * 13 \;\; 2^{81900} = 1 \;\; 81901 = 5 * 2^{14} - 19$$

$$Fourier[800 \times 600] = 901ms$$

$$Fourier[640 \times 480] = 520ms$$

$$Fourier[480 \times 320] = 290.75ms$$

$$Fourier[320 \times 240] = 122.56ms$$

$$Fourier[160 \times 120] = 27ms$$

$$Fourier[80 \times 40] = 4.15ms$$

$$!!Fourier[1024 \times 1024] = 2092ms$$

$$!Fourier[512 \times 512] = 467ms$$

$$Fourier[256 \times 256] = 91.27ms$$

$$!!Fourier[128 \times 128] = 21.17ms$$

$$Fourier[64 \times 64] = 5.13ms$$

$$Fourier[32 \times 32] = 1.42ms$$

$$Fourier[16 \times 16] = 0.5ms$$

$$!!Ferma[1024 \times 1024] = 1402ms$$

$$!Ferma[512 \times 512] = 506ms$$

$$!!Ferma[256 \times 256] = 61.12ms$$

$$Ferma[128 \times 128] = 51.05ms$$

$$Ferma[64 \times 64] = 11.41ms$$

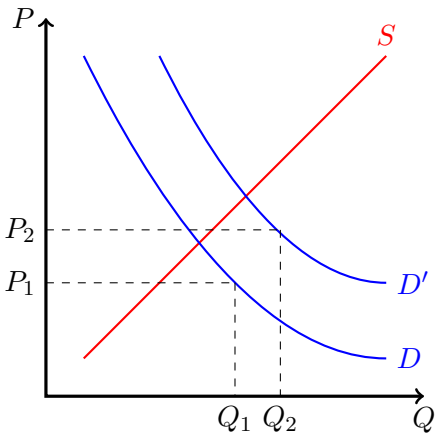
$$Ferma[32 \times 32] = 2.61ms$$

$$Ferma[16 \times 16] = 1.19ms$$

Two Dim	1024	512	256	128	64	32	16
Furie	2092	467	91.27	21.17	5.13	1.42	0.5
Ferma	1402	506	61.12	51.05	11.41	2.61	1.19

hSize	vSize	T (repeat)	Time
$256 = 16 \times 16$	$256 = 16 \times 16$	100	691
$256 = 16t \times 16t$	$256 = 16t \times 16t$	100	984
$64tt = 8 \times 8$	$64tt = 8 \times 8$	1000	500
$64t = 2 \times 32$	$64t = 2 \times 32$	1000	734

Продолжение...


P
S
P2
P1
D′
D
Q1
Q2
Q