

## ОПТИМИЗИРОВАННЫЕ АЛГОРИТМЫ ПРЕОБРАЗОВАНИЙ ХААРА ДЛЯ ОТЕЧЕСТВЕННОЙ ПЛАТФОРМЫ DSP «ELCORE» И ИХ ПРИМЕНЕНИЕ В ЗАДАЧАХ ГРАФИЧЕСКОЙ ОБРАБОТКИ ДАННЫХ

Милов А.Н.

### Введение

ГУП НПЦ «ЭЛВИС» является разработчиком собственной IP-ядерной платформы проектирования систем на кристалле «Мультикор», в состав которой входит библиотека оригинальных ядер цифровых сигнальных процессоров (DSP) серии «ELcore™». Процессоры DSP используются в решении задач таких важных областей, как радиолокация, связь, обработка, кодирование и сжатие сигналов, изображения и звука.

Процессоры серии «Мультикор» являются многоядерными микроконтроллерами с двумя типами ядер[1]: RISC-ядро является MIPS32-совместимым (R3000), выполняет функции управления и является ведущим (master). DSP-ядро «Elcore» является высокопроизводительным 32-битным ведомым (slave) устройством и имеет типичную для многих цифровых процессоров обработки сигналов гарвардскую архитектуру с внутренним параллелизмом по потокам и форматам обрабатываемых данных. Оно предназначено для высокоскоростной обработки информации в форматах с фиксированной и с плавающей точкой. Существенным достоинством данного DSP-ядра для задач обработки компьютерной графики является то, что система его инструкций обеспечивает одновременное выполнение в течение одного командного цикла до двух вычислительных операций и до двух пересылок. Достигается это наличием многопортового регистрового файла и оригинальной архитектуры арифметико-логического устройства (АЛУ).

Все вычислительные операции, поддерживаемые «Elcore», подразделяются на два типа в зависимости от операционных исполняющих устройств:

1. Операции типа OP1 - это операции, исполняемые АЛУ.
2. Операции типа OP2 – это операции, исполняемые при помощи умножителя-сдвигателя (УС).

Параллельное выполнение двух вычислительных операций DSP-ядром «Elcore» возможно только при условии, что они выполняются на разных операционных устройствах (более подробно см. [2]). DSP «Elcore» позволяет также, оставаясь в рамках одной системы инструкций, наращивать количество вычислительных секций (по SIMD-типу), увеличивая тем самым общую производительность системы.

DSP-ядро «Elcore» имеет два типа встроенной памяти, которые условно обозначаются X и Y. X-память используется как для записи, так и для чтения. Операции обмена с этой памятью могут осуществляться для любого регистра DSP. Y-память является подмножеством X-памяти и может быть использована только для чтения входных данных в регистр R0. Вся встроенная память «Elcore» является

*Рассматриваются особенности реализации одномерного и двумерного преобразований Хаара дискретных сигналов на отечественной платформе сигнальных процессоров DSP «Elcore», а также их применение для построения огрубленных версий сеток высот местности и построения списков уменьшенных образов текстур.*

двухпортовой. Это означает, что обращения к ней со стороны RISC-ядра или DMA-канала памяти происходят без приостановки конвейера «Elcore».

Преобразование Хаара является простейшим вейвлет-преобразованием, выделяющим высокочастотную (ВЧ) и низкочастотную (НЧ) составляющие сигнала. Рекурсивные формулы одномерного преобразования Хаара для сигнала  $x = \{x_j\}_{j \in \mathbb{Z}}$  могут быть записаны следующим образом [3]:

$$x_j^{(i)} = \frac{x_{2j}^{(i+1)} + x_{2j+1}^{(i+1)}}{2}, y_j^{(i)} = \frac{x_{2j}^{(i+1)} - x_{2j+1}^{(i+1)}}{2},$$

где  $i$  – уровень разрешения сигнала (для рекурсивной свертки),

$j$  – индекс соответствующего отсчета сигнала  $x$  или  $y$ ,  
 $y$  – ВЧ составляющая сигнала  $x$ .

В работе [3] было показано, что двумерное преобразование Хаара можно реализовывать не по общей схеме композиций одномерных преобразований, а по следующим формулам:

$$\begin{aligned} x_{j,k}^{(i)} &= \frac{x_{2j,2k}^{(i+1)} + x_{2j+1,2k}^{(i+1)} + x_{2j,2k+1}^{(i+1)} + x_{2j+1,2k+1}^{(i+1)}}{4}; \\ y_{j,k}^{(i)} &= \frac{x_{2j,2k}^{(i+1)} + x_{2j+1,2k}^{(i+1)} - x_{2j,2k+1}^{(i+1)} - x_{2j+1,2k+1}^{(i+1)}}{4}; \\ y_{j,k}^{\prime(i)} &= \frac{x_{2j,2k}^{(i+1)} - x_{2j+1,2k}^{(i+1)} + x_{2j,2k+1}^{(i+1)} - x_{2j+1,2k+1}^{(i+1)}}{4}; \\ y_{j,k}^{\prime\prime(i)} &= \frac{x_{2j,2k}^{(i+1)} - x_{2j+1,2k}^{(i+1)} - x_{2j,2k+1}^{(i+1)} + x_{2j+1,2k+1}^{(i+1)}}{4}. \end{aligned} \quad (1)$$

В настоящей статье представлен алгоритм реализации этих преобразований на практике и дана оценка производительности для отечественной платформы DSP «Elcore».

### Особенности реализации одномерного преобразования Хаара для «Elcore»

Предлагается следующий конвейерный алгоритм обработки данных одномерного преобразования Хаара для чисел в формате IEEE-754 (рис.1Рис.). В строках указана последовательность тактов DSP, в столбцах приведены первые шесть 32-битных регистра из регистрового файла «Elcore».

	R0	R2	R4	R6	R8	R10
1	(AT)+IT					0.5
2	(AT)+IT	R0*R10				0.5
3	(AT)+IT		R0*R10			0.5
4	(AT)+IT	R2±R4		R0*R10		0.5
5	(AT)+IT	(A0)+			R0*R10	0.5
6	(AT)+IT	R0*R10	(A1)+		R6±R8	0.5
7	(AT)+IT		R0*R10	(A0)+		0.5
8	(AT)+IT	R2±R4		R0*R10	(A1)+	0.5
9	(AT)+IT	(A0)+			R0*R10	0.5
10	(AT)+IT	R0*R10	(A1)+		R6±R8	0.5
11	(AT)+IT		R0*R10	(A0)+		0.5
12	(AT)+IT	R2±R4		R0*R10	(A1)+	0.5

Рис. 1. Реализация одномерного преобразования Хаара

Условные обозначения:

-  – операция OP1;                      0.5                      – константное значение в регистре;
-  – операция OP2;                      R0\*R10 – операция умножения регистров в формате с плавающей точкой;

R2±R4 – операция «сложение-вычитание» регистров в формате с плавающей точкой (FAS), результаты помещаются в исходные регистры;

(AT)+IT – загрузка входных данных из Y-памяти и пост-инкремент адресного регистра AT на величину IT (IT=1);

(A0)+,(A1)+ – выгрузка обработанных данных в X-память и пост-инкремент соответствующего адресного регистра (A0, A1).

Алгоритм использует две пары регистров (R2,R4) и (R6,R8) и чередует чтение, обработку и запись данных с одной пары на другую.

Входные данные для обработки загружаются из Y памяти в регистр R0. Ввиду того, что регистровый файл является многопортовым, параллельно происходит операция деления предыдущего входного отсчета, и результат помещается в регистр R2 (или R6).

Подготовка конвейера осуществляется до четвертого такта включительно. Основная стадия конвейера преобразования занимает восемь тактов (на диаграмме она показана с 5 такта по 12 включительно). На каждой такой стадии из восьми отсчетов входного сигнала образуются по четыре отсчета НЧ и ВЧ составляющих. В течение основной стадии происходит 16 обменов с памятью данных (используется два адресных генератора – AGU и AGU-Y) и 16 арифметических операций с плавающей точкой в формате IEEE-754.

Заметим, что вместо операции FAS могли быть использованы операции FADD (сложение) и FSUB (вычитание), однако в этом случае потребовался бы как минимум еще один регистр из регистрового файла. Кроме того, операция FAS позволила освободить устройство

АЛУ на 5, 9, 13 тактах. Как будет показано ниже, операция FAS наиболее эффективно может быть применена для конвейера двумерного преобразования Хаара.

Итак, производительность приведенного алгоритма ограничена выгрузкой результатов в X-память данных. Поэтому, приведенный алгоритм является полностью оптимизированным для данного устройства.

Поскольку конвейер преобразования является довольно коротким, то особо важным моментом является минимизация затрат на выполнение подготовительных действий, связанных с организацией циклов. Для этого в DSP «Elcore» предусмотрена аппаратная поддержка программных циклов типа DO, которая позволяет избежать накладных расходов при организации простых циклов, связанных с инкрементом счетчика циклов и проверкой условия конца цикла. Необходимым требованием при использовании этого аппаратного механизма является то, чтобы число повторов цикла могло быть вычислено до вхождения в цикл. Помимо этого, оно должно находиться в пределах от 1 до 16383.

Основная стадия конвейера преобразования (5-12 такты диаграммы), описанная на языке ассемблера «Elcore», приведена на рис. 2.

```

DO Nd2, LastLabel:
FMPY R0,R10,R8      R2.L,(A0)+      (AT)+IT,R0.L
FMPY R0,R10,R2      FAS R6,R8      R4.L,(A1)+      (AT)+IT,R0.L
FMPY R0,R10,R4      FAS R6,R8      R6.L,(A0)+      (AT)+IT,R0.L
FMPY R0,R10,R6      FAS R2,R6      R8.L,(A1)+      (AT)+IT,R0.L
FMPY R0,R10,R8      FAS R6,R8      R2.L,(A0)+      (AT)+IT,R0.L
FMPY R0,R10,R2      FAS R6,R8      R4.L,(A1)+      (AT)+IT,R0.L
FMPY R0,R10,R4      FAS R6,R8      R6.L,(A0)+      (AT)+IT,R0.L
LastLabel:
FMPY R0,R18,R6      FAS R4,R8      R2.L,(A0)+      (AT)+IT,R0.L
    
```

Рис.2. Алгоритм основной стадии конвейера преобразования Хаара.

## Реализация двумерного преобразования Хаара для ЦСП «Elcore»

Для того чтобы столь же эффективно реализовать двумерное преобразование Хаара, необходимо представить формулы (1) в другом виде:

$$\begin{aligned}
 x_{j,k}^{(i)} &= \left( \frac{x_{2j,2k}^{(i+1)} + x_{2j+1,2k}^{(i+1)}}{4} \right) + \left( \frac{x_{2j,2k+1}^{(i+1)} + x_{2j+1,2k+1}^{(i+1)}}{4} \right) \\
 y_{j,k}^{(i)} &= \left( \frac{x_{2j,2k}^{(i+1)} + x_{2j+1,2k}^{(i+1)}}{4} \right) - \left( \frac{x_{2j,2k+1}^{(i+1)} + x_{2j+1,2k+1}^{(i+1)}}{4} \right) \\
 y_{j,k}^{\prime(i)} &= \left( \frac{x_{2j,2k}^{(i+1)} - x_{2j+1,2k}^{(i+1)}}{4} \right) + \left( \frac{x_{2j,2k+1}^{(i+1)} - x_{2j+1,2k+1}^{(i+1)}}{4} \right) \\
 y_{j,k}^{\prime\prime(i)} &= \left( \frac{x_{2j,2k}^{(i+1)} - x_{2j+1,2k}^{(i+1)}}{4} \right) - \left( \frac{x_{2j,2k+1}^{(i+1)} - x_{2j+1,2k+1}^{(i+1)}}{4} \right)
 \end{aligned} \quad (2)$$

Заметим, что все комбинации сумм/разностей операндов в (2) можно получить с использованием всего четырех операций FAS. Обозначим:

$$\begin{aligned}
 FAS(n, m)_+ &= n + m, \\
 FAS(n, m)_- &= n - m,
 \end{aligned}$$

$$\begin{pmatrix} N \\ M \end{pmatrix} = FAS(n, m) \Leftrightarrow \begin{cases} N = FAS(n, m)_+ \\ M = FAS(n, m)_- \end{cases}$$

Тогда двумерное преобразование Хаара может быть записано следующим образом:

$$\begin{pmatrix} x_{j,k}^{(i)} \\ y_{j,k}^{(i)} \end{pmatrix} = FAS \left( FAS \left( \frac{x_{2j,2k}^{(i+1)}}{4}, \frac{x_{2j+1,2k}^{(i+1)}}{4} \right)_+, FAS \left( \frac{x_{2j,2k+1}^{(i+1)}}{4}, \frac{x_{2j+1,2k+1}^{(i+1)}}{4} \right)_+ \right)$$

$$\begin{pmatrix} y_{j,k}^{\prime(i)} \\ y_{j,k}^{\prime\prime(i)} \end{pmatrix} = FAS \left( FAS \left( \frac{x_{2j,2k}^{(i+1)}}{4}, \frac{x_{2j+1,2k}^{(i+1)}}{4} \right)_-, FAS \left( \frac{x_{2j,2k+1}^{(i+1)}}{4}, \frac{x_{2j+1,2k+1}^{(i+1)}}{4} \right)_- \right)$$

В соответствии с предложенной формой записи, алгоритм двумерного преобразования Хаара примет вид (рис. 3):

Подготовка конвейера алгоритма занимает 8 тактов, основная стадия конвейера также занимает 8 тактов, в течение которых происходит 16 обращений к памяти и 24 арифметические операции с плавающей точкой. Основная стадия соответствует паре преобразований (1), на входе которых по четыре отсчета двумерного сигнала, на выходе - также по четыре отсчета одного шага двумерного преобразования Хаара.

Заметим, что результаты преобразования могут быть выданы в различные диапазоны X-памяти. Для этого необходимо настроить соответствующие адресные регистры A0, A1, A2, A3.

Основная стадия конвейера предлагаемого алгоритма на языке ассемблера «Elcore» приведена на рис. 4.

	R0	R2	R4	R6	R8	R10	R12	R14	R16	R18	DT
1 (AT)+IT										0.25	N
2 (AT)+DT	R0*R18									0.25	N
3 (AT)-IT		R0*R18								0.25	N
4 (AT)+DT			R0*R18							0.25	N
5 (AT)+IT		R2±R4		R0*R18						0.25	N
6 (AT)+DT			R6±R8	R0*R18						0.25	N
7 (AT)-IT	R2+R6		R2-R6		R0*R18					0.25	N
8 (AT)+DT	(A0)+	R4+R8		R4-R8		R0*R18				0.25	N
9 (AT)+IT		(A1)+			R10±R12		R0*R18			0.25	N
10 (AT)+DT	R0*R18		(A2)+				R14±R16			0.25	N
11 (AT)-IT		R0*R18		(A3)+	R10+R14		R10-R14			0.25	N
12 (AT)+DT			R0*R18		(A0)+	R12+R16		R12-R16		0.25	N
13 (AT)+IT		R2±R4		R0*R18		(A1)+				0.25	N
14 (AT)+DT			R6±R8	R0*R18		(A2)+				0.25	N
15 (AT)-IT	R2+R6		R2-R6		R0*R18		(A3)+			0.25	N
16 (AT)+DT	(A0)+	R4+R8		R4-R8		R0*R18				0.25	N
17 (AT)+IT		(A1)+			R10±R12		R0*R18			0.25	N

Рис. 3. Реализация двумерного преобразования Хаара

Условные обозначения:

(AT) ± IT, (AT) ± DT – загрузка входных данных из Y-памяти и пост-инкремент (декремент) адресного регистра AT на величину регистра смещения IT или DT (IT=1, DT=N),

N – размер матрицы входного двумерного сигнала (предполагается, что матрица имеет NxN элементов).

DO Nd2, LastLabel:

```

FMPY R0, R18, R16   FAS R10, R12   R4.L, (A1)+ (AT)+DT, R0.L
FMPY R0, R18, R2    FAS R14, R16   R6.L, (A2)+ (AT)+IT, R0.L
FMPY R0, R18, R4    FAS R10, R14   R8.L, (A3)+ (AT)-DT, R0.L
FMPY R0, R18, R6    FAS R12, R16   R10.L, (A0)+ (AT)+IT, R0.L
FMPY R0, R18, R8    FAS R2, R4     R12.L, (A1)+ (AT)+DT, R0.L
FMPY R0, R18, R10   FAS R6, R8     R14.L, (A2)+ (AT)+IT, R0.L
FMPY R0, R18, R12   FAS R2, R6     R16.L, (A3)+ (AT)-DT, R0.L
LastLabel:
FMPY R0, R18, R14   FAS R4, R8     R2.L, (A0)+ (AT)+IT, R0.L

```

Рис. 4. Основная стадия алгоритма двумерного преобразования Хаара

Как видно из рисунка, на каждом такте происходит по два обмена с памятью данных, одна операция умножения и одна операция «сложение-вычитание». Поскольку выходные данные могут быть записаны только в X-память, предложенный алгоритм также является полностью оптимизированным для данного устройства.

### Оценка производительности предложенных алгоритмов

Архитектура операционных блоков DSP-ядер серии «Elcore» включает два режима обработки данных:

- «Один поток инструкций, один поток данных» (Single Instruction Single Data - SISD);
- «Один поток инструкций, множественные потоки данных» (Single Instruction Multiple Data - SIMD).

Отличие SIMD-режима от SISD состоит в том, что за один командный цикл одна арифметическая инструкция выполняется в нескольких одинаковых операционных блоках одновременно (а именно, наборах АЛУ и УС), причем с разными данными. Таким образом, применение режима SIMD в DSP разработки НПЦ «ЭЛВИС» способно ускорить исполнение математической задачи в несколько раз. DSP-ядро «Elcore-24» микросхемы 1892BM2T «MC-24» обладает двумя операционными SIMD-секциями, то есть вычисления в нем могут быть ускорены в два раза.

Преобразование Хаара локально, т.е. затрагивает только ту область сигнала, к которой оно применяется. Поэтому каждый шаг преобразования Хаара любого одномерного или двумерного сигнала можно осуществить, разложив исходный сигнал на несколько составляющих. Например, если необходимо осуществить один шаг двумерного преобразования Хаара для сигнала, заданного двумерной матрицей размером 128x128, то такое преобразование эквивалентно четырем преобразованиям Хаара для составляющих сигнала с размерами 64x64 или двум преобразованиям для составляющих сигнала с размерами 128x64 или 64x128.

Таким образом, производительность приведенных алгоритмов преобразования Хаара может быть увеличена в два раза включением SIMD-режима и разложением исходного сигнала на составляющие.

В таблице 1 представлены характеристики производительности предложенных алгоритмов для различных

DSP-ядер серии «Elcore». Ввиду того, что внутренняя память DSP-ядер реализована непосредственно на кристалле (а, следовательно, существенно ограничена), в таблице также представлены граничные размеры входных сигналов, для которых возможно преобразование без разделения на составляющие.

Для достижения указанной пиковой производительности при обработке непрерывных потоков данных обычно используют блочную обработку. Суть её состоит в разделении сигнала на составляющие блоки такого размера, чтобы одновременно в памяти входных данных DSP могло содержаться два таких блока. Таким образом, в то время, пока DSP производит обработку одного из блоков, DMA-канал производит выгрузку обработанных данных предыдущего и загрузку входных данных следующего блока.

Для эффективного разделения двумерных сигналов на составляющие и последующей их загрузки в Y-память в микроконтроллерах серии «Мультикор» предусмотрен режим двумерной DMA-передачи. Таким образом, нет необходимости в предварительном построении двумерных массивов данных или организации серии DMA-передач.

### Применение преобразования Хаара

Преобразования Хаара применяются для построения списков уменьшенных копий текстур с размерами  $2^n \times 2^n, n \in N$  (рис. 5) при осуществлении пространственно-аккуратного текстурирования (англ. тер. – mip-mapping) [4].

В работе [3] было показано, что преобразование Хаара может также применяться для сжатия сеток высот поверхностей, параметризуемых на плоскости. На рис. 6 приведены результаты применения одного шага двумерного преобразования Хаара к заданной сетке высот. Слева направо, сверху вниз: монотонные изображения сеток высот размерами 64x64 и 32x32 соответственно, перспективное представление поверхности, заданной сеткой высот, триангуляция исходной сетки высот 64x64, триангуляция сетки высот 32x32, полученной с помощью двумерного преобразования Хаара. Все изображения получены с использованием графической библиотеки с интерфейсом OpenGL для микропроцессоров серии «Мультикор».

Таблица 1.

Характеристики предложенных алгоритмов.

Преобразование	DSP-ядро	Ограничения вх. сигналов, отсчетов	Пиковая производительность, тактов на входной отсчет
Одномерное в формате IEEE-754	Elcore-14(SISD)	12 288	1
	Elcore-24(SIMD)	8 192	0.5
Двумерное в формате IEEE-754	Elcore-14(SISD)	12 100 (110x110)	1
	Elcore-24(SIMD)	8 100 (90x90)	0.5
Двумерное в формате R8G8B8X8 <sup>1</sup>	Elcore-14(SISD)	12 100 (110x110)	2
	Elcore-24(SIMD)	8 100 (90x90)	1

<sup>1</sup> Широко используется для обработки растровых изображений и текстур. Известно так же как 32-битный формат растровых изображений.



Рис. 5. Список уменьшенных копий текстур

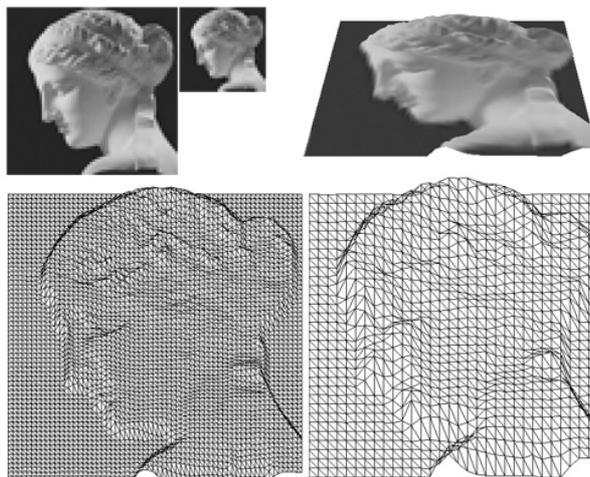
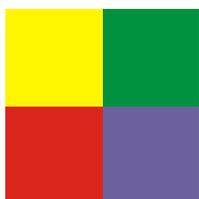


Рис. 6. Один шаг двумерного преобразования Хаара для заданной сетки высот

#### Литература

1. «Микросхема интегральная 1892ВМЗТ. Руководство пользователя», ГУП НПЦ «ЭЛВИС», Москва, 2005 г.
2. «DSP-ядро ELcore-x4. Система инструкций», ГУП НПЦ «ЭЛВИС», Москва, 2006 г.
3. Переберин А.В., диссертация кандидата физико-математических наук «Многомасштабные методы синтеза и анализа изображений», ИГМ им. Келдыша, Москва, 2002 г.
4. Lance Williams, «Pyramidal parametrics», Computer graphics, volume 17, number 3, 1983.



**MATROX**  
IMAGING

Компания Matrox Imaging более 30 лет успешно поставляет свои ПК-ориентированные программные и аппаратные технологии разработчикам *систем машинного зрения, анализа объектов и изображений, цифрового видео наблюдения, систем визуализации в медицине и биологии, систем управления технологическими процессами.*

Линейка продуктов включает в себя: *фрейм-грабберы, процессоры обработки изображения, видео смарт-камеры, готовые системы машинного зрения, программное обеспечение для анализа изображения (библиотеки MIL). Штаб квартира в Монреале (Канада), офисы и представительства в 20 странах мира.*

Авторизованный представитель компании **Matrox Imaging** в России – **System Video Graphics Alliance (SVGA).**

## КРАТКИЕ СООБЩЕНИЯ

УДК 621.372

## ЭФФЕКТИВНЫЕ ОКОННЫЕ ФУНКЦИИ НА ОСНОВЕ ФУНКЦИЙ ГАУССА-ЭРМИТА

Вашкевич М.И.

Рассматривается метод расчета эффективных оконных функций, используемых при гармоническом анализе сигналов с применением ДПФ.

Как известно, задача минимизации мощности спектра оконных функций на заданном частотном интервале приводит к использованию вытянутых сфероидальных волновых функций (ВСВФ). Особенность их в том, что они являются собственными (инвариантными) функциями преобразования Фурье на ограниченном интервале. Это означает, что преобразование Фурье переводит каждую ВСВФ в такую же функцию с точностью до константы. Значит, задав во временной области достаточно «узкую» функцию, можно получить функцию ограниченную и в частотной области. Одной из наиболее известных и удачных аппроксимаций ВСВФ являются оконные функции Кайзера. Кроме ВСВФ известен ряд функций, инвариантных к преобразованию Фурье в бесконечных пределах. Функции Гаусса-Эрмита, как один из примеров таких функций, имеют вид:

$$\varphi_n = \omega_n(x) \cdot e^{-x^2/2},$$

где  $\omega_n(x)$  - многочлен Эрмита  $n$ -го порядка.

Из сказанного выше можно сделать вывод, что функции Гаусса-Эрмита (ФГЭ) могут являться неплохим «материалом» для построения оконных функций. Тем более, что в [1] было показано, что энергия ФГЭ сконцентрирована на ограниченном отрезке, как во временной, так и в частотной области.

Окно, которое исследовалось в данной работе, имеет вид:

$$\omega(k) = e^{\frac{(k/N)^2}{2\sigma^2}} \times \frac{H_4\left(\frac{k \cdot \alpha}{N}\right)}{H_4(0)}, \quad k \in \left[-\frac{N}{2}; \frac{N}{2}\right],$$

$N$  – половина ширины окна,

$H_4(x)$  - многочлен Эрмита 4-го порядка.

В этой формуле между  $\alpha$  и  $\sigma$  существует определенная эмпирически установленная зависимость:  $\sigma(\alpha) = 0.96275 - 0.8375 \cdot \alpha$ . Параметр  $\alpha$  может принимать значения из интервала  $[0; 0.5]$ . По сути  $\alpha$  - это константа, аналогичная константе  $\beta$  в окне Кайзера. При увеличении  $\alpha$  происходит расширение главного лепестка, но при этом падает уровень боковых лепестков.

Сравнение данного окна производилось с окном Кайзера в среде MatLab 6.5 (при помощи утилиты wvtool). Параметр  $\beta$  в окне Кайзера подбирался таким образом, чтобы ширина главного лепестка была одинаковой у обоих окон. Во всех случаях длина окна  $N=65$ . В таблице 1 приведены наиболее интересные из полученных результатов.

Таблица 1.

Сравнение экспериментального окна с окном Кайзера

Экспериментальное окно			Окно Кайзера			разность уровней боковых лепестков (дБ)
$\alpha$	$K_y, \%$	УБЛ (дБ)	$\beta$	$K_y, \%$	УБЛ (дБ)	
0,30	0,67	25,3	2,91	0,70	23,6	1,7
0,35	0,33	29,0	3,34	0,35	26,3	2,7
0,40	0,11	34,3	3,95	0,13	30,2	4,1
0,45	0,03	39,6	4,72	0,03	35,4	4,2
0,50	0,01	44,6	5,76	0,01	42,3	2,3

Здесь:  $K_y$  - коэффициент утечки, который показывает, какая доля общей мощности сосредоточена в боковых лепестках его спектра. УБЛ - уровень боковых лепестков (уровень максимального из боковых лепестков относительно главного лепестка).

Как видно из полученных результатов, предлагаемое окно имеет несколько лучшие показатели по сравнению с окном Кайзера (при  $\beta = [3; 6]$ ), и поэтому может успешно применяться при гармоническом анализе сигналов с применением ДПФ.

## Литература

1. Хонина С.Н., Котляр В.В. Влияние дифракции на изображения, согласованные с вытянутыми сфероидальными волновыми функциями // Компьютерная оптика, ИСОИ РАН, Самара-Москва, 2001, №21, с. 64-69.