

ОПТИМИЗАЦИЯ СТРУКТУРНЫХ И ВЫЧИСЛИТЕЛЬНЫХ ХАРАКТЕРИСТИК АЛГОРИТМОВ ВОССТАНОВЛЕНИЯ БОЛЬШЕФОРМАТНЫХ ИЗОБРАЖЕНИЙ

Н.В. Радченко, В.В. Сергеев, В.М. Чернов, М.А. Чичева
Институт систем обработки изображений РАН
Самарский государственный аэрокосмический университет

Аннотация

В работе решается задача восстановления большеформатных изображений посредством фильтрации. Описывается метод секционированной свертки. Представляется метод выбора оптимального размера блока. Приводится базовый алгоритм ДПФ для выбранного размера блока, указывается его вычислительная сложность, а также характеристики реализации.

Введение

В настоящей работе решается задача восстановления (или фильтрации) большеформатных изображений с целью устранения искажений, которые появляются в процессе формирования и передачи изображения по каналам связи. Существует множество работ, в которых решаются подобные задачи (см., например, [1], [2], [3]). Однако, даже описанный в литературе алгоритм секционированной свертки [1], который может быть применен к большеформатным изображениям, как правило, не предусматривает оптимизации алгоритма по вычислительной сложности.

Есть хорошо известные программы обработки изображений, позволяющие выполнять фильтрацию (например, Adobe Photoshop). Однако такие программы, как правило, не предназначены для обработки большеформатных изображений, и, тем более, для фильтрации изображения по мере поступления информации. Так попытка простого «открытия» в Adobe Photoshop 220-мегабайтного изображения в формате TIFF требует около 300 Мб дискового пространства и выполняется около 9 минут на компьютере Pentium-III, 650МГц, 128Мб RAM.

Ниже будут изложены основы разработанного авторами подхода, позволяющего оптимизировать вычислительную сложность фильтрации.

Процедура восстановления и ее оптимизация

Вычисление свертки спектральным методом

Известно, что выходная последовательность фильтра представляет в математических терминах свертку исходной последовательности $x(n)$ и импульсной характеристики фильтра $h(n)$. В свою очередь, известный способ эффективного вычисления свертки состоит в использовании теоремы о свертке и быстрого преобразования Фурье (FFT) [1].

Пусть $y(m) = (x * h)(m)$ - дискретная круговая свертка двух N -периодических функций:

$$y(m) = \sum_{n=0}^{N-1} x(n)h(m-n). \quad (1)$$

Тогда, как известно, для спектров дискретных преобразований Фурье

$$\hat{y}(m) = \sum_{n=0}^{N-1} y(n) \omega^{mn}, \quad (2)$$

где $\omega = \exp\left\{2\pi i / N\right\}$, $m=0, 1, \dots, N-1$, справедливо соотношение

$$\hat{y}(m) = \hat{x}(m) \hat{h}(m). \quad (3)$$

Поэтому вычисление свертки (1) с помощью (2) и (3) может быть проведено по схеме приведенной на рис. 1, где ОДПФ - обратное дискретное преобразование Фурье:

$$y(n) = N^{-1} \sum_{m=0}^{N-1} \hat{y}(m) \omega^{-mn}. \quad (4)$$

Наличие быстрых алгоритмов вычисления преобразований (2) и (4) позволяет существенно снизить вычислительные затраты.

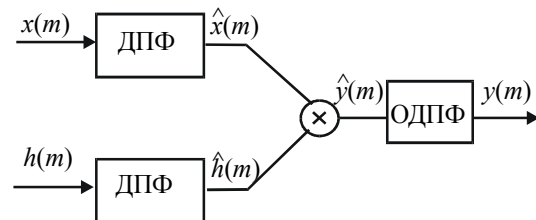


Рис. 1. Вычисление циклической свертки через БПФ

Очевидно, что суммарная сложность такого вычисления свертки определяется сложностью FFT и равна

$$W(N) = 2W_{FFT}(N) + \lambda, \quad (5)$$

где $W_{FFT}(N)$ - сложность преобразования Фурье, λ - количество дополнительных операций для вычисления произведения спектров. Здесь предполагается, что сложность вычисления прямого и обратного преобразования Фурье одинакова, и спектр $\hat{h}(m)$ импульсной характеристики фильтра $h(n)$ рассчитан заранее.

Секционированное вычисление свертки

Для задач линейной цифровой фильтрации сигналов, т.е. вычисления дискретной аperiodической свертки

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k), \quad (6)$$

типичной является ситуация, когда длина преобразуемой последовательности $x(n)$ во много раз превышает ширину конечной импульсной характеристики $h(k)$. В этом случае эффективным является метод секционированного вычисления свертки. Произведем анализ вычислительной сложности и оптимизацию процедуры вычисления свертки с

секционированием. Следует отметить, что известны два основных метода секционирования свертки: перекрытия с суммированием и перекрытия с накоплением [2]. Второй метод несколько эффективнее по объему вычислений и удобнее в программной реализации, поэтому ниже будет рассматриваться только он.

Пусть импульсная характеристика (ядро свертки) имеет длину в M отсчетов и отлична от нуля на интервале $[-M^+, M^-]$ ($M^+ + M^- + 1 = M$). Тогда пределы суммирования в (6) становятся конечными:

$$y(n) = \sum_{k=-M^-}^{M^+} h(k)x(n-k). \quad (7)$$

При вычислении одномерной аperiodической свертки вида (7) использование секционирования по методу перекрытия с накоплением заключается в следующем. Из входной последовательности выделяются блоки $x^{(l)}(k)$ размером в N отсчетов ($N \geq M$) с перекрытием в $(M-1)$ точках (см. рис. 2):

$$x^{(l)}(k) = x[k+l(N-M+1)]$$

при $0 \leq k \leq N-1, l=0, \pm 1, \pm 2, \dots$

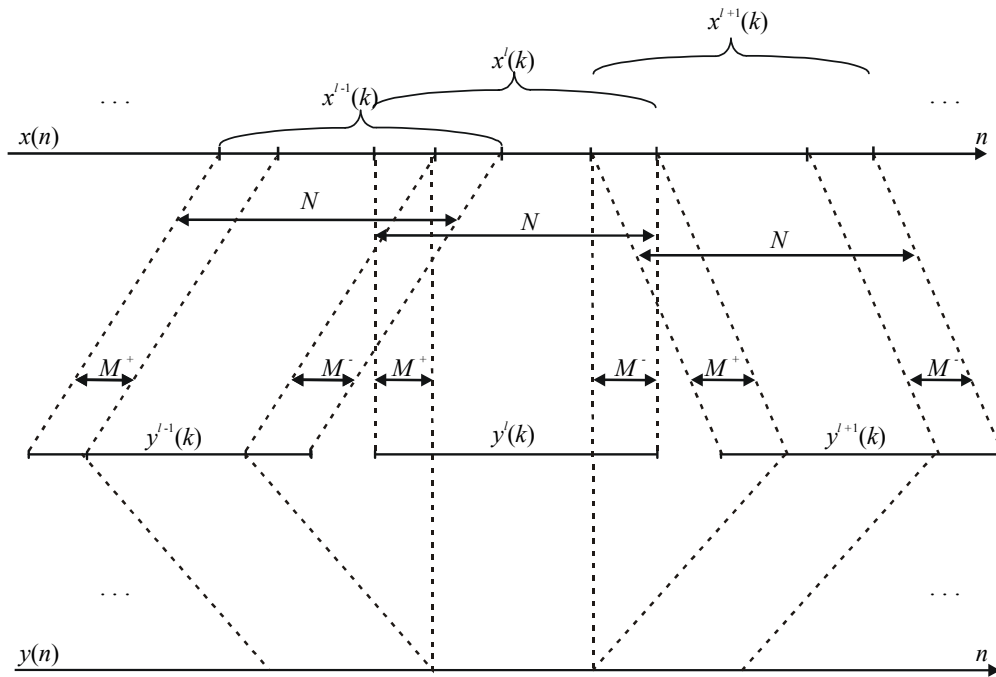


Рис. 2. Схема расположения блоков при секционировании одномерной свертки методом перекрытия с накоплением

С помощью дискретного ортогонального преобразования длины N для каждого блока вычисляется его циклическая свертка с импульсной характеристикой (дополненной до этой длины нулями), в результате чего формируются выходные блоки $y^{(l)}(k)$ такой же длины. Поскольку импульсная характеристика конечна, искажениям из-за эффектов наложения вследствие цикличности подвержены лишь крайние отсчеты выходных блоков (по M^+ и M^- с каждой стороны), а их средние $(N-M+1)$ отсчетов точно совпадают с искомыми элементами аperiodической свертки. Поэтому заключительным шагом преобразований является получение выходной последовательности из средних частей выходных блоков:

$$y[k+l(N-M+1)] = y^{(l)}(k),$$

$$M^+ + 1 \leq k \leq N - M^- - 1.$$

Таким образом, на каждые $(N-M+1)$ отсчетов выходной последовательности вычисляется одна циклическая свертка, что требует выполнения $W(N)$ арифметических операций (5). Считая входную последовательность бесконечной, определим число операций, приходящееся на один отсчет выходного сигнала. Для импульсной характеристики длиной M оно равно:

$$W(M) = \frac{W(N)}{N-M+1}. \quad (8)$$

Величина (8) служит критерием оптимизации спектрального алгоритма вычисления свертки с секционированием и должна быть минимизирована за счет:

а) выбора наиболее быстрого алгоритма ДОП длины N , используемого при вычислении циклической свертки;

б) выбора оптимальной длины секции N .

Результаты такой оптимизации представлены ниже.

Данный подход легко обобщается на случай линейной фильтрации двумерных сигналов (изображений), т.е. вычисления двумерной свертки

$$\begin{aligned} y(n_1, n_2) &= \\ &= \sum_{k_1=-M_1^-}^{M_1^+} \sum_{k_2=-M_2^-}^{M_2^+} h(k_1, k_2)x(n_1-k_1, n_2-k_2), \end{aligned}$$

где $M_1^+, M_2^+, M_1^-, M_2^-$ - границы прямоугольной области ненулевых отсчетов двумерной импульсной характеристики. Двумерное поле отсчетов изображения, которое считается бесконечным (заданным

на всей плоскости аргументов) разбивается на равные прямоугольные блоки размерами $N_1 \times N_2$ с шагами $(N_1 - M_1 + 1)$ и $(N_2 - M_2 + 1)$, где $M_1 = M_1^- + M_1^+ + 1$, $M_2 = M_2^- + M_2^+ + 1$:

$$x^{(l_1, l_2)}(k_1, k_2) = x[k_1 + l_1(N_1 - M_1 + 1), k_2 + l_2(N_2 - M_2 + 1)]$$

при $0 \leq k_1 < N_1$, $0 \leq k_2 < N_2$, $l_1, l_2 = 0, \pm 1, \pm 2, K$

С помощью двумерного дискретного ортогонального преобразования размерами $N_1 \times N_2$ для каждого блока вычисляется его циклическая свертка с импульсной характеристикой, в результате чего формируются выходные блоки $y^{(l_1, l_2)}(k_1, k_2)$ таких же размеров. Далее из средних частей выходных блоков формируется искомым результат:

$$y[k_1 + l_1(N_1 - M_1 + 1), k_2 + l_2(N_2 - M_2 + 1)] = y^{(l_1, l_2)}(k_1, k_2)$$

при $M_1^+ + 1 \leq k_1 \leq N_1 - M_1^- - 1$, $M_2^+ + 1 \leq k_2 \leq N_2 - M_2^- - 1$

Аналог критерия (8), т.е. число арифметических операций, приходящееся на один отсчет выходного сигнала, в данном случае задается выражением:

$$U(M_1, M_2) = \frac{W(N_1, N_2)}{(N_1 - M_1 + 1)(N_2 - M_2 + 1)},$$

где $W(N^1, N^2)$ - число операций при вычислении двумерной циклической свертки размерами $N^1 \times N^2$.

Оптимизация спектрального алгоритма для вычисления свертки с секционированием

Наиболее популярным и известным алгоритмом вычисления ДПФ (2) до сих пор остается алгоритм быстрого преобразования Фурье (БПФ), предложенный в 1965г. Кули и Тьюки [1] и рассматриваемый большинством пользователей для последовательностей, длина (или период) которых равна целой степени двух: $N=2^r$. Неблагоприятным следствием популярности БПФ-алгоритма Кули-Тьюки является широкое распространение мнения о том, что применять дискретное преобразование Фурье практически лишь при такой длине последовательности. В результате БПФ-алгоритмы стали диктовать параметры применяемых устройств вместо того, чтобы приложения диктовали выбор подходящего алгоритма БПФ [1].

В задачах, не предполагающих жестких аппаратных требований к длине обрабатываемого сигнала, применение только традиционных алгоритмов БПФ Кули-Тьюки приводит к необходимости увеличивать период обрабатываемых сигналов - добавлять нулевые отсчеты до ближайшего «хорошего» целого числа вида $N^* = 2^r$. Такие числа расположены в натуральном ряду весьма редко, что приводит к почти двукратному увеличению удельной вычислительной сложности для «плохих» N , равных, например, $2^r + 1 = N$ (см. рис.3 график 1).

На самом деле быстрые алгоритмы преобразования Фурье могут быть построены для большого

набора длин. В настоящей работе авторы производят выбор оптимальной структуры алгоритма длины N на основе, следующих БПФ-алгоритмов:

- 1) БПФ Кули-Тьюки с декомпозицией по основанию 2 [1], [2];
- 2) БПФ Кули-Тьюки с декомпозицией по основаниям 3 и 6 [4];
- 3) редукция Гуда-Томаса, для формирования алгоритмов промежуточных длин [1], [2];
- 4) специальные алгоритмы ДПФ коротких длин [2];
- 5) использование идеи совмещения для вещественных входных данных [1], [2].

На рис. 3 представлены графики удельной арифметической сложности вычисления спектра комплексной одномерной последовательности произвольной длины. Здесь N - реальный период; $W^*(N)$ - удельная арифметическая сложность (число арифметических операций на один отсчет входного сигнала) вычисления БПФ сигнала с увеличенным периодом $N^* > N$. На графике 1 приведена сложность вычисления спектра традиционным способом - на базе БПФ Кули-Тьюки по основанию 2.

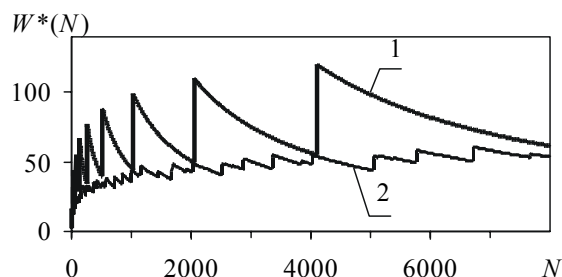


Рис. 3. Удельная сложность вычисления одномерного спектра

На графике 2 представлена вычислительная сложность наилучшего алгоритма, выбранного из предложенного набора. При этом сложность преобразования уменьшается в среднем в 1,6 раза. Отметим, что для ряда длин удельная сложность вычисления спектра снижается более чем в два раза.

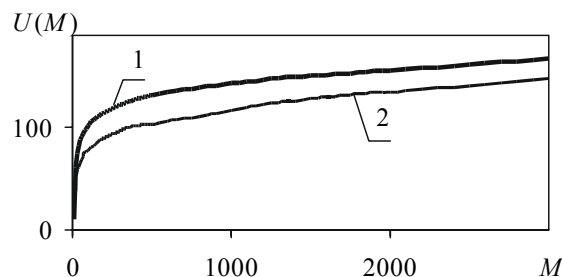


Рис. 4. Удельная сложность вычисления одномерной свертки

Рис. 4 показывает зависимость удельной сложности вычисления секционированной свертки длинного сигнала от размера импульсной характеристики. Здесь также первый график соответствует использованию традиционного БПФ Кули-Тьюки по основанию 2, а второй график показывает удельную сложность вычисления свертки при использовании предлагаемых алгоритмов. Сочетание секциониро-

вания и эффективных БА ДПФ приводит к снижению сложности вычисления свертки в 1,1-1,4 раза.

В двумерном случае появляются дополнительные возможности для снижения вычислительной сложности БПФ, а, следовательно, и свертки (фильтрации). В частности, заметный эффект дает применение существенно двумерных алгоритмов, вместо известной процедуры построчно-столбцового преобразования.

На рис. 5 приведена минимальная удельная сложность вычисления спектра двумерного массива объема $N \times N$ посредством преобразования Фурье объема $N^* \times N^*$, где $N^* > N$ выбрано из предлагаемого диапазона длин. График 1 построен для случая, когда используется построчно-столбцовый способ преобразования при $N^* = 2^r$, график 2 показывает возможности построчно-столбцовых алгоритмов из расширенного набора длин, график 3 построен при использовании существенно двумерных алгоритмов.

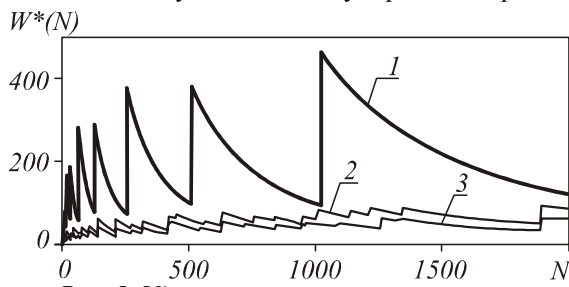


Рис. 5. Удельная сложность вычисления двумерного спектра

Исследования показывают, что в двумерном случае расширение диапазона длин позволяет уменьшить вычислительную сложность в среднем на 63,8% (или до 92%, то есть практически в 2 раза в отдельных случаях). Переход к существенно двумерным алгоритмам позволяет увеличить этот выигрыш в среднем до 74% (или до 97,5% в отдельных случаях).

Аналогичные результаты могут быть представлены по результатам оптимизации выбора размера блока для двумерной секционированной свертки.

Оптимальная реализация

быстрого двумерного алгоритма ДПФ 216×216

Задача оптимизации спектрального алгоритма для вычисления свертки с секционированием была решена нами для использования в конкретной системе обработки большеформатных изображений с заданными размерами выходного блока не менее 170×170 пикселей и размерами импульсной характеристики 45×19 пикселей. В этих условиях оптимальным оказалось использование БПФ 216×216 (или $6^3 \times 6^3$), алгоритм которого предлагается ниже.

Представление комплексных чисел в виде γ -кодов

Пусть

$$\gamma = \exp\left(\frac{2\pi i}{3}\right) = \frac{1}{2}(-1 + i\sqrt{3}), \quad \bar{\gamma} = \frac{1}{2}(-1 - i\sqrt{3}).$$

Тогда для комплексного числа z , наряду с алгебраической формой представления $z = a + bi$, возможна и форма $z = \gamma s + \bar{\gamma} p$ [4], где вещественные x и

y связаны с числами a и b соотношениями

$$s = \left(\frac{b/\sqrt{3}}{\sqrt{3}}\right) - a, \quad p = \left(-\frac{b/\sqrt{3}}{\sqrt{3}}\right) - a.$$

(В частности, для вещественных z справедливы равенства $z = -s = -p$).

Пару вещественных чисел (s, p) , ассоциированную с представлением z в виде $z = \gamma s + \bar{\gamma} p$, будем называть γ -кодом числа z . Арифметические действия над комплексными числами индуцируют следующие правила действия над кодами:

$$(s, p) + (u, v) = (s + u, p + v), \quad (9)$$

$$(s, p) \cdot (u, v) = ((p - s)(v - u) + sv, (p - s)(v - u) - pv)$$

Таким образом, сложение комплексных чисел, представленных γ -кодами, реализуется с помощью двух вещественных сложений, а умножение - с помощью трех вещественных умножений и трех сложений, то есть не сложнее, чем умножение и сложение комплексных чисел в обычной алгебраической форме. Особенно просто реализуется умножение на числа γ и $\bar{\gamma}$, имеющие соответственно коды $(1, 0)$ и $(0, 1)$. Соотношение (9) в этом случае принимает вид:

$$\begin{aligned} (1, 0) \cdot (u, v) &= (u, u - v) \\ (0, 1) \cdot (u, v) &= (v - u, -u) \end{aligned} \quad (10)$$

и не содержит нетривиальных умножений.

Отметим, что использование такого представления данных выгодно не только для алгоритмов преобразования Фурье по основанию 3 [4], но и для БПФ-алгоритмов по основанию 6, так как корень шестой степени из единицы равен $-\gamma$:

$$\exp\left\{\frac{2\pi i}{6}\right\} = \exp\left\{\frac{\pi i}{3}\right\} = -\gamma.$$

Описание алгоритма

Пусть $N = 216$ и $x(n_1, n_2)$ - двумерная входная комплексная последовательность, $\omega = \exp\left(\frac{2\pi i}{N}\right)$, $\hat{x}(m_1, m_2)$ - двумерный дискретный спектр Фурье:

$$\hat{x}(m_1, m_2) = \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} x(n_1, n_2) \omega^{m_1 n_1 + m_2 n_2}. \quad (11)$$

Тогда редукция преобразований (11) к ДПФ массивов объема $\left(\frac{N}{6} \times \frac{N}{6}\right)$ осуществляется по формуле:

$$\begin{aligned} \hat{x}(m_1, m_2) &= \sum_{a,b=0}^5 \omega^{am_1 + bm_2} \sum_{n_1, n_2=0}^{N/6-1} x(6n_1 + a, 6n_2 + b) \times \\ &\times (\omega^6)^{m_1 n_1 + m_2 n_2} = \sum_{a,b=0}^5 S_{a,b}(m_1, m_2) \end{aligned} \quad (12)$$

т.е. вычисление ДПФ объема 216×216 сводится к 36 ДПФ объема 36×36 и т.д. При этом уже на втором шаге, для вычисления ДПФ объема 6×6 при представлении степеней ω и входной последовательности γ -кодами уже не требуется вещественных умножений.

Еще одним фактором, упрощающим алгоритм ДПФ, является то, что функции $S_{a,b}(m_1, m_2)$ достаточно вычислять для пар (m_1, m_2) , лежащих в относительно малой "фундаментальной области":

$$\Lambda = \left\{ 0 \leq m_1, m_2 \leq \frac{N}{6} - 1 \right\}.$$

Действительно, если $\tau = \exp\left(\frac{\pi i}{3}\right)$ и $(m_1, m_2) \in \Lambda$,

то

$$S_{a,b}\left(m_1 + \alpha \frac{N}{6}, m_2 + \beta \frac{N}{6}\right) = \tau^{a\alpha + b\beta} S_{a,b}(m_1, m_2).$$

Реализация предложенного алгоритма ДПФ предполагает выполнение следующей последовательности шагов.

Шаг 1. Формирование массива γ -представлений входной двумерной последовательности размерами 216×216 , а также констант алгоритма (степеней ω), если последние не сформированы заранее.

Шаги 2-4. Последовательное выполнение трех шагов вычислений согласно формуле редукции (12), и в результате - формирование двумерного дискретного Фурье-спектра, представленного в γ -кодах.

Заметим также, что описанный алгоритм позволяет вычислить не только прямое, но и обратное преобразование. Для ОДПФ шаги 1-4 должны выполняться в обратном порядке: сначала из дискретного спектра, представленного в γ -кодах, получается γ -представление выходной двумерной последовательности, а затем выполняется операция декодирования, т.е. перевода последовательности в традиционную комплексную форму.

Кроме того, для перемножения спектров при вычислении циклической свертки блока нет необходимости в переходе от гамма-кодов к стандартному комплексному представлению, так как умножение комплексных чисел в обеих формах представления имеет одинаковую сложность.

Вычислительная сложность и структурная характеристика алгоритма. Анализ вычислительной сложности предложенного алгоритма ДПФ показывает, что он требует выполнения около 50 арифметических операций в расчете на один комплексный пиксел преобразуемого двумерного массива 216×216 и является наиболее простым в реализации среди всех известных алгоритмов ДПФ массива $(N \times N)$ при $150 \leq N \leq 360$.

Для сравнения отметим, что традиционные строочно-столбцовые быстрые алгоритмы двумерного ДПФ в указанном диапазоне размеров имеют вычислительную сложность не менее 80 операций на пиксел.

Кроме того, существенным достоинством алгоритма является его простая структура.

Схема декомпозиции однородна, включает всего три шага. Вычисления (шаги 2-4) могут реализовываться как в "прямой" форме (от одноточечных преобразований к 6-точечным, 6²-точечным и т.д.), так и с помощью обратной рекурсии.

Экспериментальное исследование

Разработанный алгоритм ДПФ 216×216 используется теперь в программном комплексе обработки и восстановления больших изображений. Время восстановления (фильтрации, свертки) полутонового изображения 15000×15000 пикселей (около 220 Мб) на компьютере Pentium-III, 650 МГц, 128 Мб RAM составляет 5 минут. При замене эффективного алгоритма ДПФ 216×216 традиционным алгоритмом преобразования Фурье наиболее близкого размера 256×256 время обработки изображения возрастает до 8,5 минут, то есть в 1,7 раза.

Заключение

Таким образом, в работе представлена методика оптимизации восстановления большеформатных изображений по критерию вычислительной сложности. Показаны способы формирования эффективных алгоритмов одно- и двумерных БПФ в широком диапазоне длин. Разработанные алгоритмы эффективнее традиционного БПФ в 1,5-2 раза. Для программного комплекса обработки изображений при заданных параметрах разработан алгоритм БПФ 216×216 с представлением данных в γ -кодах, являющийся наиболее эффективным в заданном диапазоне длин. Экспериментально подтверждена эффективность его использования.

Работа выполнена при финансовой поддержке Российского Фонда Фундаментальных Исследований, проект № 00-01-00600.

Литература

1. R.E. Blahut *Fast algorithms for digital signal processing* // Addison-Wesley Publishing Company, Inc., 1985.
2. H.G. Nussbaumer *Fast Fourier transform and convolution algorithms* // Springer-Verlag Berlin Heidelberg, 1981.
3. B. Aramberpola, P.J.W. Rayner *Multidimensional fast Fourier transform algorithm* // *Electron. Lett.* 1979. 15(3).
4. V.M. Chernov *Algorithms, realizable in Hamilton-Eisenstein codes, for two-dimensional discrete orthogonal transforms* // *Problems Inform. Transmission*, 1995. Vol. 31. N. 3. P. 228-235.