

# ОБРАБОТКА ИЗОБРАЖЕНИЙ, РАСПОЗНАВАНИЕ ОБРАЗОВ

## ИССЛЕДОВАНИЕ ПРИВЕДЕННОГО КОМПЕТЕНТНОГО АЛГОРИТМА НАД МНОЖЕСТВОМ АЛГОРИТМОВ ВЫЧИСЛЕНИЯ СВЕРТКИ

А.Ю. Баврина, В.В. Мясников

Институт систем обработки изображений РАН,

Самарский государственный аэрокосмический университет им. С.П. Королева

### Аннотация

В работе исследуются операция приведения и операция построения приведенного компетентного алгоритма над множеством известных алгоритмов постоянной сложности. В качестве известных алгоритмов опорного множества используются алгоритм прямого вычисления свертки и наиболее известные алгоритмы на основе быстрых дискретных ортогональных преобразований (с декомпозицией Кули-Тьюки и Гуда-Томаса, алгоритм Рейдера для коротких длин). Показано, что совместное их использование, которое дает приведенный компетентный алгоритм, позволяет снизить вычислительную сложность формируемого алгоритма вычисления свертки даже по отношению к наилучшим алгоритмам опорного множества.

### Введение

Рассмотрим задачу вычисления свертки:

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

Здесь  $\{x(n)\}_{n=0}^{N-1}$  – входной сигнал,  $\{h(m)\}_{m=0}^{M-1}$  – конечная импульсная характеристика,  $\{y(n)\}_{n=0}^{N-M+1}$  – выходной сигнал. Обозначим задачу (1) через  $Z = Z(\mathfrak{S}_0, \{x(n)\}_{n=0}^{N-1})$ , где  $\mathfrak{S}_0 = (\{h(n)\}_{n=0}^{M-1}, (N, \emptyset))$  – априорная информация о задаче.

### Компетентный алгоритм

Введем функцию:

$$\text{par} \left( Z \left( \mathfrak{S}_0, \{x(n)\}_{n=0}^{N-1} \right) \Big|_{\mathfrak{S}_0 = (\{h(n)\}_{n=0}^{M-1}, (N, \mathfrak{S}_{(1)}))} \right) = (M, N).$$

**Определение 1** Алгоритм  $A$  называется *алгоритмом постоянной сложности*, если для любой задачи  $Z$  выражение для его вычислительной сложности имеет вид аналитической функции, зависящей только от величины  $\text{par}(Z)$ , то есть:

$$U(A(Z)) \equiv u_A(\text{par}(Z)) = u_A(M, N).$$

Обозначим  $\mathfrak{K}_{A(M,N)}$  область определения алгоритма постоянной сложности.

**Определение 2** *Компетентным алгоритмом*  $A^\oplus \notin \{A\}$  над опорным множеством  $\{A\}$  алгоритмов постоянной сложности называется алгоритм вида:

$$\mathfrak{K}_{A^\oplus(M,N)} = \bigcup_{A \in \{A\}} \mathfrak{K}_{A(M,N)}, \quad \forall Z \in \mathfrak{K}_{A^\oplus(M,N)}$$

$$A(Z) \equiv A^*(Z) \quad \text{if} \quad \left( \begin{array}{l} (M, N) = \text{par}(Z) \wedge \\ u_{A^*}(M, N) = \\ = \min_{\substack{A \in \{A\} \\ (M,N) \in \mathfrak{K}_{A(M,N)}}} u_A(M, N) \end{array} \right).$$

Очевидно, компетентный алгоритм для решения конкретной задачи вычисления свертки использует тот алгоритм из опорного множества, сложность которого при решении данной задачи минимальна.

**Предложение 1** [1] Компетентный алгоритм  $A^\oplus$  над множеством алгоритмов постоянной сложности  $\{A\}$  является алгоритмом постоянной сложности.

### Приведенный алгоритм

**Определение 3** Функция сложности  $u_A(M, N)$  алгоритма  $A$  постоянной сложности называется *корректной*, если выполняется неравенство:

$$\forall (M, N) \in \mathfrak{K}_{A(M,N)} \quad u_A(M, N) \leq$$

$$\left( \begin{array}{l} \min_{\substack{m=1, M-2 \\ (m, N-(M-m)) \in \mathfrak{K}_{A(M,N)}, \\ (M-m, N-m) \in \mathfrak{K}_{A(M,N)}}} \left( \begin{array}{l} u_A(m, N-(M-m))+ \\ u_A(M-m, N-m)+ \\ \xi_{\text{add}} \end{array} \right), \\ \min_{\substack{m=0, 1, 2, \dots \\ n=0, 1, 2, \dots \\ (M+m, N+m+n) \in \mathfrak{K}_{A(M,N)}}} \left( \begin{array}{l} u_A(M+m, N+m+n) \cdot \\ (N-M+n+1) \\ (N-M+1) \end{array} \right), \\ \min_{\substack{n=1, \lfloor N/2 \rfloor \\ (M,n) \in \mathfrak{K}_{A(M,N)}, \\ (M, \\ (N-n+M-1)) \in \mathfrak{K}_{A(M,N)}}} \left( \begin{array}{l} u_A(M, n)(n-M+1) + \\ u_A(M, N-n+M-1)(N-n) \\ N-M+1 \end{array} \right) \end{array} \right).$$

**Определение 4** Алгоритм постоянной сложности с корректной функцией сложности называется *приведенным*.

**Теорема 1** [1] Для любого алгоритма постоянной сложности  $A$ , заданного на  $\mathfrak{K}_{A(M,N)}$ , существует (может быть сконструирован) приведенный алгоритм  $\tilde{A}$ :

$$\begin{aligned} \mathfrak{K}_{\tilde{A}(M,N)} &= \mathfrak{K}_{A(M,N)}, \\ \forall (M,N) \in \mathfrak{K}_{A(M,N)} \quad u_{\tilde{A}(M,N)} &\leq u_{A(M,N)}. \end{aligned}$$

**Теорема 2** [1] Пусть  $\{A\}$  – множество алгоритмов постоянной сложности, из которых построено множество  $\{\tilde{A}\}$  приведенных алгоритмов постоянной сложности. Пусть  $\tilde{A}_1^\oplus$  – приведенный компетентный алгоритм над множеством  $\{A\}$  с областью определения  $\mathfrak{K}_{A_1^\oplus}$ , а  $\tilde{A}_2^\oplus$  – приведенный компетентный алгоритм над множеством  $\{\tilde{A}\}$  с областью определения  $\mathfrak{K}_{A_2^\oplus}$ . Тогда выполняются соотношения:

$$\begin{aligned} \mathfrak{K}_{\tilde{A}_1^\oplus} &= \mathfrak{K}_{A_2^\oplus} (\equiv \mathfrak{K}_{A_1^\oplus}), \\ \forall Z \in \mathfrak{K}_{\tilde{A}_1^\oplus} \quad U(\tilde{A}_1^\oplus(Z)) &= U(\tilde{A}_2^\oplus(Z)). \end{aligned}$$

#### Исследование приведенного компетентного алгоритма

Исследование проводилось для наиболее известных алгоритмов вычисления свертки, формирующих множество:

$$\{A_{DC}, A_2, A_{2_S}, A_3, A_4, A_{GT}\}. \quad (2)$$

Здесь:

- $A_{DC}$  – алгоритм прямого вычисления свертки [2] согласно формуле (1);
- $A_2, A_3, A_4$  – вычисления свертки с использованием декомпозиции Кули-Тьюки по основаниям «2», «3» и «4» соответственно (используется дополнение входного сигнала нулями до степени соответствующего основания) [3];
- $A_{2_S}$  – с декомпозицией Кули-Тьюки по основанию «2», дополнением нулями до степени «2» и секционированием [3, 4];
- $A_{GT}$  – с декомпозицией Гуда-Томаса и алгоритма Рейдера для вычисления малоточечных ДПФ (с использованием дополнения входного сигнала нулями до длины, раскладывающейся на взаимно простые сомножители) [3].

Исследование проводилось в двух направлениях.

*Первое направление* – это определение влияния операции приведения на вычислительную слож-

ность алгоритмов быстрой свертки. В том числе на те, в которых используется секционирование входного сигнала. Результаты исследования представлены на рисунках 1 и 2 и таблице. Поскольку функция сложности  $u_{\cdot}(M,N)$  зависит от двух координат  $M$  и  $N$ , на рисунках представлены только «сечения» этой функции для значения  $M = 60$ .

По результатам этого направления исследований можно сделать следующие *выводы*:

- функция сложности алгоритмов вычисления свертки на основе ДОП и с секционированием, и без секционирования не является корректной;
- использование операции приведения для алгоритмов вычисления свертки, основанных на ДОП с декомпозицией Кули-Тьюки по различным основаниям с секционированием сигнала и без него, позволяет *уменьшить сложность решения большинства задач (50-56%)* в области определения соответствующих алгоритмов (рис.1, 2);
- *значительное снижение сложности* для указанных алгоритмов происходит для задач с параметрами  $(M,N)$ , у которых длина сигнала  $N$  чуть больше степени основания декомпозиции Кули-Тьюки (рис.1, 2). Для таких задач возможно снижение сложности *более чем на порядок* (показатель (2б) в таблице).

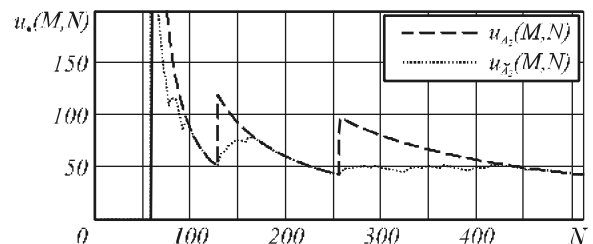


Рис. 1. Функция сложности после операции приведения

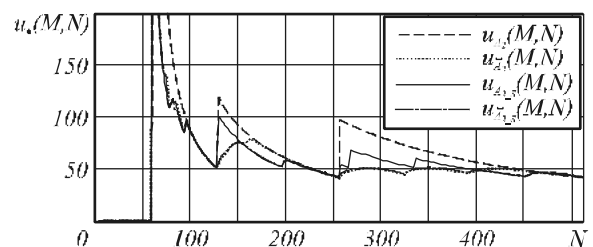


Рис. 2. Функция сложности после операции приведения для алгоритма с секционированием

*Вторым направлением* является исследование приведенного компетентного алгоритма. Для этого из указанного в (2) множества алгоритмов постоянной сложности выбиралось некоторое подмножество  $A$ , называемое опорным. Сложность каждого алгоритма опорного множества определялась на области  $\mathfrak{K}_{(M,N)} = [1, 512] \times [1, 512]$ . Над опорным подмножеством  $A$  строился компетентный алгоритм  $A^\oplus$ . Для него выполнялась операция приведения, результатом которой являлся приведенный компетентный алгоритм  $\tilde{A}^\oplus$ . На рисунках 3-5 (по мере

увеличения числа алгоритмов опорного множества) приведены функции сложности для алгоритмов опорного множества и построенного над ними приведенного компетентного алгоритма.

По представленным результатам можно сделать следующие *выводы*:

- независимо от числа алгоритмов опорного множества, можно рассчитывать на снижение сложности в относительно большом (более 30%) числе отсчетов области определения;
- для отсчетов области определения, в которых произошли изменения функции сложности в результате операции приведения, можно рассчитывать на снижение сложности в среднем на 10-30% (от сложности компетентного алгоритма).
- как число отсчетов области определения, в которых операция приведения снижает сложность компетентного алгоритма, так и собственно величина этого снижения зависит не только (а с учетом представленных результатов, не столько) от числа алгоритмов опорного множества, а в значительной степени от вида функции сложности этих алгоритмов. То есть, в общем случае нет монотонного поведения ни одного показателя от числа алгоритмов опорного множества (таблица), кроме средней сложности итогового алгоритма  $\tilde{A}^{\oplus}$ ;
- по мере наращивания мощности опорного множества алгоритмов, величина *снижения средней сложности* и компетентного, и приведенного компетентного алгоритма оказывается все меньше (см. поведение показателей (5) и (6) в таблице). В частности, при переходе от трех к пяти алгоритмам в опорном множестве показатель (6) снижается всего на 10% (с 113 до 102 операций) по сравнению с почти 30% снижения (со 165 до 117) при переходе от одного алгоритма к двум. Поэтому на практике представляется малоэффективным процесс бесконечного наращивания мощности опорного множества алгоритмов. Практически тех же результатов (по сложности решения задач свертки) можно добиться и при сравнительно малом количестве алгоритмов. По

той же причине, и процесс синтеза новых алгоритмов вычисления свертки, *которые по показателю сложности не превосходят радикально существующие*, с практической точки зрения является нецелесообразным.

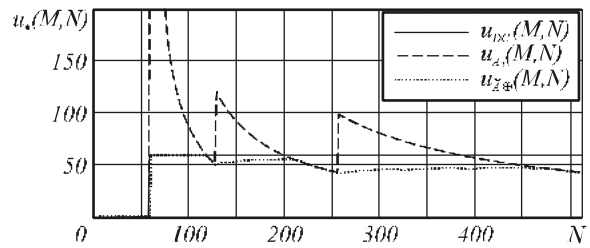


Рис. 3. Функция сложности для приведенного компетентного алгоритма,  $\mathbf{A} = \{A_{DC}, A_2\}$

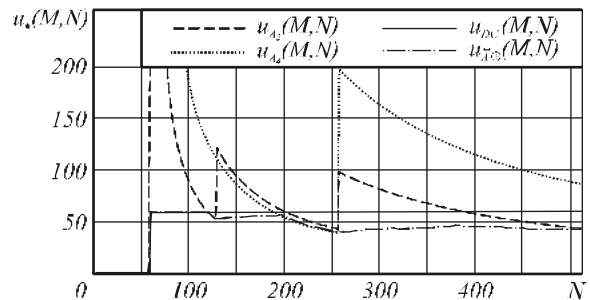


Рис. 4. Функция сложности для приведенного компетентного алгоритма,  $\mathbf{A} = \{A_{DC}, A_2, A_4\}$

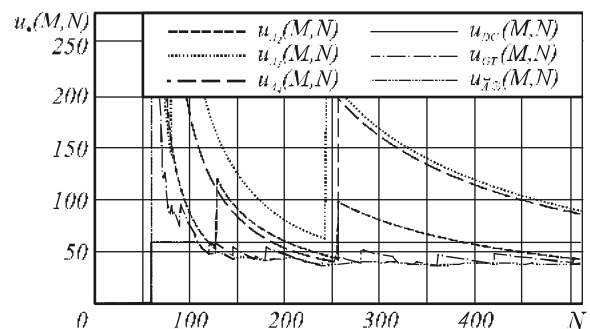


Рис. 5. Функция сложности для приведенного компетентного алгоритма,  $\mathbf{A} = \{A_{DC}, A_2, A_3, A_4, A_{GT}\}$

Таблица 1. Сводные характеристики изменения сложности алгоритмов решения задачи Z при использовании операции приведения

Показатель	Алгоритм				
	$A_{2_s}$	$A_2$	$\{A_{DC}, A_2\}^{\oplus}$	$\left\{ \begin{matrix} A_{DC}, A_2, \\ A_4 \end{matrix} \right\}^{\oplus}$	$\left\{ \begin{matrix} A_{DC}, A_2, A_3, \\ A_4, A_{GT} \end{matrix} \right\}^{\oplus}$
(1) относительное число отсчетов области определения, в которых произошло снижение сложности алгоритма	0.499	0.563	0.348	0.394	0.423
(2а) максимальное абсолютное уменьшение сложности алгоритма	17863	17863	77.717	88.478	91.944
(2б) максимальное относительное уменьшение сложности алгоритма	12.102	12.603	1.929	2.132	1.323

(3а) среднее абсолютное уменьшение сложности алгоритма, нормированное на количество отсчетов, в которых произошло изменение сложности	271.070	250.367	24.353	29.870	10.932
(3б) среднее относительное уменьшение сложности алгоритма, нормированное на количество отсчетов, в которых произошло изменение сложности	1.405	1.650	1.220	1.274	1.104
(4а) среднее абсолютное уменьшение сложности алгоритма, нормированное на мощность области определения	135.160	140.902	8.4693	11.754	4.620
(4б) среднее относительное уменьшение сложности алгоритма, нормированное на мощность области определения	0.701	0.928	0.424	0.501	0.466
(5) средняя сложность компетентного алгоритма	299.505	306.115	125.414	124.831	106.618
(6) средняя сложность приведенного алгоритма	164.346	165.213	116.944	113.0767	101.998

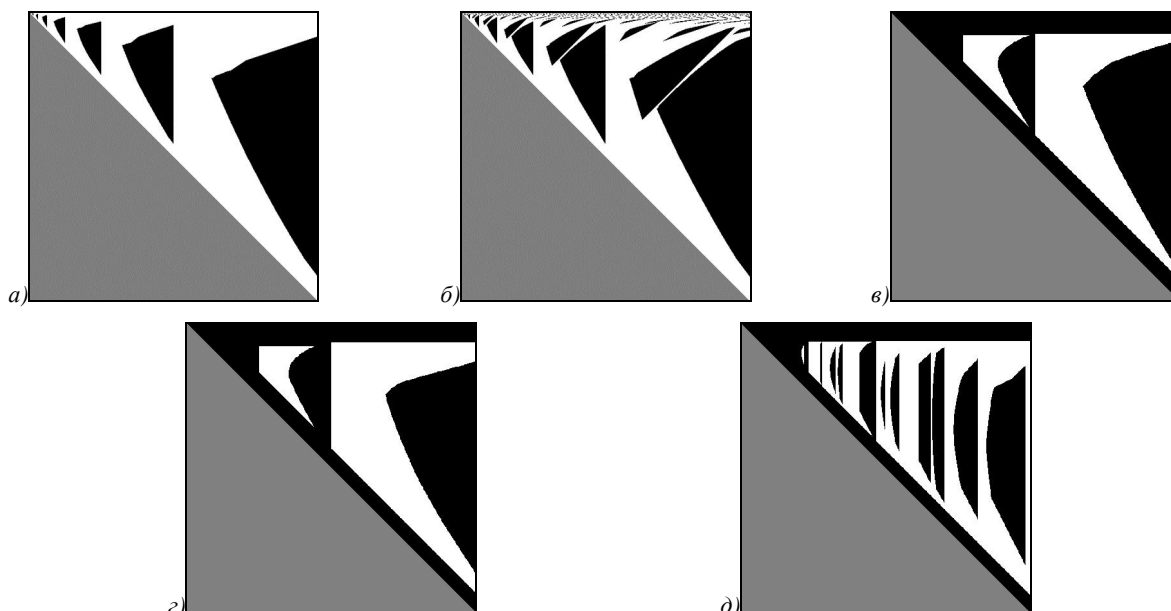


Рис. 6. Отсчеты области определения, в которых операция приведения снизила сложность соответствующего алгоритма решения задачи вычисления свертки: (серым - отсчеты, не входящие в область определения, белым - отсчеты области определения, в которых операция приведения снизила сложность соответствующего алгоритма, черным - отсчеты области определения, в которых операция приведения не изменила сложность соответствующего алгоритма)

а)  $A_2$ ; б)  $A_{2-s}$ ; в)  $\{A_{DC}, A_2\}^{\oplus}$ ; г)  $\{A_{DC}, A_2, A_4\}^{\oplus}$ ; д)  $\{A_{DC}, A_2, A_3, A_4, A_{GT}\}^{\oplus}$

### Благодарности

Работа выполнена при поддержке:

- РФФИ (проекты № 06-01-00616-а, № 07-07-97610-р\_офи);
- Фонда содействия отечественной науке;
- российско-американской программы «Фундаментальные исследования и высшее образование» (CRDF Project RUX0-014-SA-06).

### Литература

- [1] **Мясников В.В.** О синтезе эффективного алгоритма над множеством алгоритмов вычисления свертки // Компьютерная оптика, 2006. В. 29. С. 78-117.
- [2] **L.R. Rabiner, B. Gold** Theory and applications of digital signal processing // New Jersey: Prentice-Hall, Inc. Englewood Cliffs, 1975.
- [3] **H.J. Nussbaumer** Fast Fourier Transform and Convolution Algorithms // Heidelberg, Germany: Springer, 1990. P. 276.
- [4] **R.E. Blahut** Fast Algorithms for Digital Signal Processing // Reading, MA: Addison-Wesley Inc., 1984.

# STUDY OF THE REDUCED COMPETENT ALGORITHM COMPARED TO A SET OF CONVOLUTION ALGORITHMS

A.Yu. Bavrina<sup>1,2</sup>, V.V. Myasnikov<sup>1,2</sup>

<sup>1</sup>Image Processing Systems Institute of the RAS,

<sup>2</sup>Samara State Aerospace University

## Abstract

Reduction and build-up operations for the reduced competent algorithm compared to a set of existing sophisticated algorithms are investigated. As actual reference-set algorithms, we use a direct convolution algorithm and the best-known algorithms based on discrete fast orthogonal transformations (with Cooley-Tukey and Good-Thomas decomposition, Raider algorithm for high-frequency waves.) It is shown that their joint use provided by the reduced competent algorithm may lower computational complexity of the generated convolution algorithm with respect to reference-set algorithms.

**Keywords:** convolution, computational complexity, fast orthogonal transformations, reduced competent algorithm.

**Acknowledgements:** The work was supported by:  
the Russian Foundation for Basic Research (projects No. 06-01-00616-a, No. 07-07-97610-r\_ofi);

the Russian Science Support Foundation;

and the Russian-American Basic Research and Higher Education Program (CRDF Project RUX0-014-SA-06).

**Citation:** Bavrina AYu, Myasnikov VV. Study of the reduced competent algorithm compared to a set of convolution algorithms [In Russian]. Computer Optics 2008; 32(1): 64-67.

## References

- [1] Myasnikov VV. On the synthesis of the efficient algorithm over the set of the convolution algorithms [In Russian]. Computer Optics 2006; 29: 78-117.
- [2] Rabiner LRB. Gold Theory and applications of digital signal processing. New Jersey: Prentice-Hall, Inc. Englewood Cliffs 1975.
- [3] Nussbaumer HJ. Fast Fourier Transform and Convolution Algorithms. Heidelberg, Germany: Springer 1990; 276 pp.
- [4] Blahut RE. Fast Algorithms for Digital Signal Processing. Reading, MA: Addison-Wesley Inc. 1984.