Для цитирования: Радиопромышленность. — 2016. — № 1. — С. 33—37.
УДК 621.396.967

А.В. Ксендзук[1], П.А. Герасимов[1]

[1]ПАО «МАК “Вымпел”»

ИНВЕРСНЫЙ СИНТЕЗ АПЕРТУРЫ В НЕИЗЛУЧАЮЩИХ РЛС

В работе рассмотрена оптимизация цифровой обработки при инверсном синтезе апертуры в неизлучающей радиолокационной системе. Представлены алгоритмы вычисления и выполнен сравнительный анализ скорости вычисления на центральном и графическом процессоре.

Ключевые слова: *инверсный синтез апертуры, неизлучающая радиолокационная система, технология CUDA, цифровое наземное телевидение.*

Введение

В мобильных неизлучающих радиолокационных системах (НРЛС), выполняющих обнаружение и определение координат воздушных целей по сигналам сторонних передатчиков (станций телевизионного и радиовещания, базовых станций сотовой связи и пр.), классическими методами невозможно получить высокое разрешение в азимутальной и угломестной плоскостях. Это связано, прежде всего, с малым размером приемных антенн и диапазоном частот сторонних передатчиков.

Идентификация целей, обнаруживаемых неизлучающими РЛС, как следует из экспериментальных работ [1-2], может быть выполнена по оценкам их ЭПР, высоты и составляющих вектора скорости, по спектру отраженного сигнала. Дополнительная информация в виде радиолокационного изображения (РЛИ) цели в азимутальной и/или угломестной плоскости повысит достоверность идентификации. Получить такие РЛИ можно за счет применения метода инверсного (обратного) синтезирования апертуры антенны.

Реализация данного метода требует высокопроизводительной вычислительной системы. Перспективной технологией повышения производительности вычислительной системы является использование графического процессора.

Графический процессор (GPU) за счет параллельной архитектуры построения, состоящей из сотен или тысячи процессоров (ядер), позволяет распараллелить задачу и повысить производительность вычислений до 10 ТФлопс. Наибольшая эффективность достигается при реализации алгоритмов, содержащих однотипные математические вычисления над большими массивами данных без логических ветвлений. К таким операциям относятся поэлементное перемножение векторов с последующим суммированием элементов, умножение матриц и др. Такие операции используются в алгоритмах цифровой корреляционной обработки сигналов и уже показали высокую эффективность при вычислении двухмерной функции неопределенности в реальном масштабе времени [3]. В [4] показана возможность распараллеливания задачи между видеокартами, при этом эффективность распараллеливания оказалась близка к максимально возможной.

Метод обратного (инверсного) синтезирования апертуры

Метод обратного (инверсного) синтезирования основывается на корреляционной обработке с опорным сигналом, учитывающим движение цели. Такая обработка сигнала движущейся цели в неподвижной РЛС эквивалентна наблюдению неподвижной цели РЛС с синтезированной апертурой, вид которой совпадает с трассой движения цели (рисунок 1).

Потенциально обратный (инверсный) синтез позволяет получить разрешение, соответствующее половине длины антенны наземной станции. Для ис-

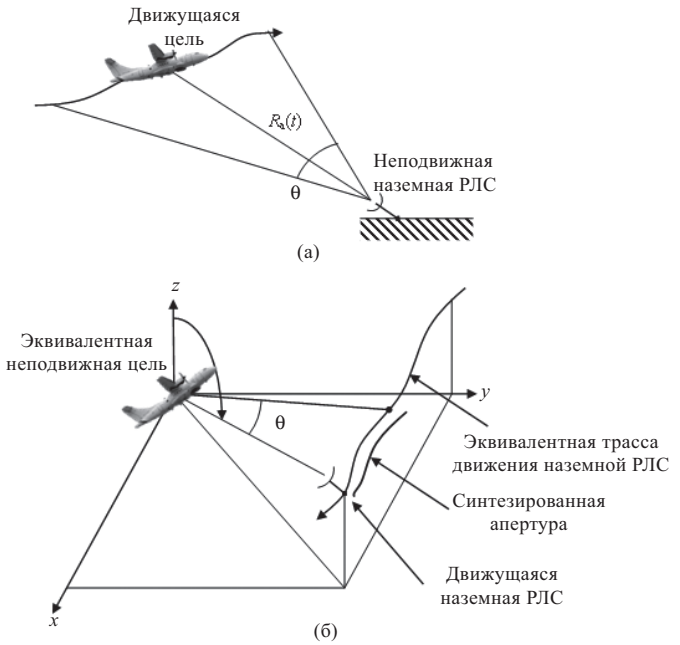

Рисунок 1. Принцип обратного синтеза апертуры

пользуемых в неизлучающих РЛС-антеннах можно получить разрешение около 20—50 см.

При обратном (инверсном) синтезе в качестве опорного используется сигнал, учитывающий структуру отражений от цели, определяемую изменением расстояния передатчик—цель—приемник $\tau\{r_A(t)\}$:

$$S[t,\tau\{r_A(t)\}] = S_0[t-\tau\{r_A(t)\}]\exp\left[-j2\pi f_0(t-\tau\{r_A(t)\})\right], \quad (1)$$

где $S[t,\tau\{r_A(t)\}]$ — отраженный от цели сигнал с комплексной огибающей $S_0[t-\tau\{r_A(t)\}]$ и несущей частотой f_0.

При классической обработке, описанной в [5, 6] в опорном сигнале учитываются фиксированные время задержки $\tau(r_A)$ и сдвиг частоты $F(r_A)$:

$$Y_c[r_A] = \int_0^T S[t,\tau\{r_A(t)\}]S^*_{оп}[t,\tau(r_A),F(r_A)]dt, \quad (2)$$

при обратном синтезе апертуры — сигнал $S[t,\tau\{r_A(t)\}]$, задержка, сдвиг частоты и фаза которого определяются изменением положения цели:

$$Y_{снс}[r_A(t)] = \int_0^{T_c} S[t,\tau\{r_A(t)\}]S^*_{оп}[t,\tau\{r_A(t)\}]dt. \quad (3)$$

Очевидно, что для обработки в реальном масштабе времени в условиях априорной неопределенности необходимо формировать матрицу опорных сигналов (1), что существенно увеличивает требования к вычислителю. Исключение составляет построение радиолокационных изображений космических объектов, так как их параметры движения могут быть предсказаны с высокой точностью.

В неизлучающих РЛС обнаружения воздушных целей обратный (инверсный) синтез апертуры необходимо выполнять в режиме постобработки. При этом в реальном времени выполняются операции обнаружения и сопровождения целей; сигнал, поступающий с приемной антенны, записывается в файл. После выхода цели из диаграммы направленности антенны выполняется инверсный синтез (3), где в качестве опорного используется матрица сигналов (1), учитывающая погрешности оценки координат и скорости целей. Учет этой информации позволяет существенно сократить требования к вычислителю и выполнить обратное синтезирование сразу после выхода цели из диаграммы направленности, а при динамически улучшающемся разрешении РЛИ — в процессе наблюдения цели.

Для сокращения времени обработки и требований к вычислителю необходимо оптимизировать алгоритм (3) для получения радиолокационного изображения цели. Эта оптимизация сводится к максимально быстрому вычислению корреляционного интеграла.

Анализ методов вычисления алгоритма

Опорный сигнал в неизлучающей радиолокационной системе формируется из принятого сигнала прямого распространения:

$$S^*_{оп}[t,\tau\{r_A(t)\}] = S^*_{пр}[t-\tau\{r_A(t,r_{A0},V_r,a_r)\}]\cdot exp\left[-j2\pi f_d(t,V_r,a_r)t\right], \quad (4)$$

где $r_A(t,r_{A0},V_r,a_r)$ — расстояние, пройденное сигналом на пути передатчик—цель—приемник; $f_d(t,V_r,a_r)$ — частота Доплера, учитывающая искомые параметры; r_{A0} — расстояние в начальный момент времени синтеза; V_r — радиальная скорость ЛА относительно радиолокатора; a_r — радиальное ускорение ЛА относительно радиолокатора.

Расстояние r_A и частота Доплера f_d при аппроксимации движения цели равноускоренным определяются выражениями:

$$r_A(t,r_{A0},V_r,a_r) = r_{A0} + V_r t + a_r\frac{t^2}{2},$$

$$f_d(t,V_r,a_r) = \frac{1}{\lambda}\frac{d}{dt}r_A(t) = \frac{1}{\lambda}(V_r + a_r t)..$$

Тогда опорный сигнал (4) зависит от трех переменных r_{A0},V_r,a_r:

$$S_{оп}[t,r_{A0},V_r,a_r] = S_{оп}[t-\tau\{t,r_{A0},V_r,a_r\}]\cdot ex\,p[-j2\pi f_d(t,V_r,a_r)t]. \quad (5)$$

Принятые сигналы после АЦП описываются значениями в дискретные моменты времени — $S[i]$, $S_{op}[i]$, где индекс $i = 0 \ldots N-1$ связан со временем t частотой дискретизации Fs: $t = i\cdot Fs^{-1}$. Значения времени задержки $\tau\{r_A(t,r_{A0},V_r,a_r)\}$ будут представлены дискретными значениями $m = \tau\{t,r_{A0},V_r,a_r\}\cdot Fs$. Тогда уравнение (3) для дискретных сигналов примет вид:

$$Y[m] = max_{V_r,a_r}\|W[r_A(m),V_r,a_r]\|^2,$$

$$W[r_A(m),V_r,a_r] = \sum_{i=0}^{N-1} S[i]\cdot\left\{S_{пр}[i-m]\cdot exp\left[-j2\pi\frac{f_d(i,V_r,a_r)}{Fs}i\right]\right\}^*. \quad (6)$$

Вычисление $S^*_{пр}[i-m]$ может быть выполнено методом округления $\tau\{t,r_{A0},V_r,a_r\}\cdot Fs$ до целого значения или методом интерполяции и последующей децимации сигнала $S_{пр}[i]$.

Вычисление $exp\{-j2\pi f_d(i,V_r,a_r)Fs\cdot i\}$ может быть выполнено заранее и сохранено в памяти устройства в виде массива $EXP[i,V_r,a_r]$. Тогда

$$W[r_A(m),V_r,a_r] = \sum_{i=0}^{N-1} S[i]\cdot\left\{S_{пр}[i-m]\cdot EXP[i,V_r,a_r]\right\}^*. \quad (7)$$

Рассчитать (7) можно методом прямой свертки (ПС) в независимых циклах по V_r,a_r:

$$W(m,V_r) = \sum_{i=0}^{N-1} S[i-m]\cdot U^*[i,V_r], \quad (8)$$

где $U[i,V_r,a_r] = S_{пр}[i]\cdot EXP[i,V_r,a_r]$.

Для этого потребуется 12*NLva* арифметических операций с комплексными числами, где *L* — количество значений *m*; *v* — количество значений V_r; *a* — количество значений a_r. Для упрощения переменную a_r выведем из дальнейшего рассмотрения.

Существует как минимум 2 алгоритма повышения скорости вычисления.

Первый алгоритм основывается на представлении (7) в виде:

$$W(m,V_r)=\sum_{i=0}^{N-1}M[i,m]\cdot exp\left(-j2\pi\frac{f_d(V_r)}{Fs}i\right), \quad (9)$$

где $M[i,m]=S[i]\cdot S^*_{\text{пр}}[i,m]$.

Алгоритм (8) эквивалентен вычислению дискретного преобразования Фурье (ДПФ) от $M[i,m]$. Данный алгоритм требует выполнения цикла по дискретному времени задержки m. В результате выполнения одного цикла вычисляются значения для частот от –*Fs*/2 до +*Fs*/2.

Реализация быстрого ДПФ (ДБПФ) требует $3N/2log_2N$ арифметических операций с комплексными числами [7]. Тогда для вычисления алгоритма (9) потребуется $L(6N+3N/2log_2N)$ арифметических операций.

Второй алгоритм заключается в вычислении (8) методом быстрой свертки (БС):

$$W(m,V_r)=FFT^{-1}\left\{FFT\{S[i]\}\cdot\left\{FFT(S_{\text{пр}}[i,R_0,V_r])\right\}^*\right\}, \quad (10)$$

где *FFT* и FFT^{-1} — обозначение прямого и обратного преобразования Фурье.

Алгоритм (10) требует выполнения цикла по радиальной скорости V_r. В результате выполнения одного цикла вычисляются дискреты времени задержи от 0 до *N*–1. Следовательно, реализация потребует $V_r(12N+9N/2log_2N)$ арифметических операций.

Оптимизация алгоритмов обратного синтеза

В методе быстрой свертки в цикле по переменной V_r выполняется БПФ отраженного сигнала, не зависящего от V_r. Вынеся данную операцию за цикл, получим количество операций $V_r(12N+3N/2log_2N)+3N/2log_2v$.

При обратном синтезе необходимо формирование массива $EXP[i,V_r]$. При большом времени накопления ($N>2^{20}$ отсчетов) и широком диапазоне радиальных скоростей V_r сформированный массив $EXP[i,V_r]$ может достигать объема более 2 ГБ. При линейном распределении искомых значений V_r через равный интервал $V_r[k]-V_r[k+1]=dV_r=const$ вычисление $U[i,V_r]$ можно представить в виде:

$$U_v[i]=S_{\text{пр}}[i]\cdot exp\left(-j2\pi\frac{V_{rmin}}{Fs\lambda}i\right)\cdot exp\left(-j2\pi\frac{dV_r\cdot v}{Fs\lambda}i\right). \quad (11)$$

Процедура перебора искомых значений V_r заключается в умножении опорного сигнала на массив $EXP_{V_{rmin}}[i]$ и последующего циклического умножения результата на $EXP_{dV_r}[i]=exp\left(-j2\pi\frac{dV_r}{Fs\lambda}i\right)$.

Реализация алгоритмов на центральном процессоре

На центральном процессоре (CPU) был реализован метод свертки и быстрой свертки. При реализации метода быстрой свертки использовалась открытая библиотека FFTW [8] для вычисления ДБПФ. Вычисления производились на одном ядре процессора.

Реализация алгоритмов на графическом процессоре

На GPU были реализованы метод быстрой свертки и реализация метода свертки в матричном исполнении.

Метод свертки в матричном исполнении описывается выражением:

$$W(m,V_r)=S_\tau[m,k]*U^*_m[k,p], \quad (12)$$

где *k* — индекс времени, *k*=0… *N*+*L*–1; *p* — индекс радиальной скорости V_r[p], *p*=0…*v*–1; $S_\tau[m,k]=\begin{cases}S[k-m], \text{если } m\le k\le m+N\\ 0, \text{ в остальных случаях}\end{cases}$; $Um[k,p]=$ $=S_{\text{пр}}[k]\cdot exp\left(-j2\pi\frac{V_r[p]}{Fs\lambda}k\right)$.

На GPU была выполнена операция перемножения матриц с использованием открытой библиотеки cuBLAS. Формирование матриц S_τ и U_mосуществлялось на CPU. Реализация данного метода потребует $6(N+L)Lv$ арифметических операций.

Реализации алгоритма на GPU имеют свои особенности, сказывающиеся на времени вычисления. Время вычислений на GPU состоит из трех слагаемых: $t_{GPU}=t_1+t_2+t_3$, где t_1 — время, необходимое для передачи массивов из оперативной памяти в глобальную память GPU; t_2 — время, затраченное на выполнение операций над массивами; t_3 — время, требуемое для передачи результатов вычисления из глобальной памяти GPU в оперативную память.

Время, необходимое для передачи данных и требуемое на передачу данных из глобальной памяти GPU в оперативную память, определяется пропускной способностью шины PCI Express.

Сравнение скорости вычислений на CPU и GPU

Выполним сравнительный анализ времен вычисления методов ПС и БС, реализованных на CPU и методов БС и умножения матриц на GPU при следующих параметрах обработки:

- количество отсчетов по времени $N=2^h$, где *h*=[14…24];
- количество отсчетов по времени задержки *L*=1000;
- количество отсчетов по радиальной скорости *v*=500.

Реализация данных методов потребует производительности при *h*=24 для метода ПС — 100,7 ТФлопс, БС — 403 ГФлопс (при времени вычисления 1 с).

Параметры используемого оборудования представлены в таблице 1.

Программы, реализующие представленные методы, были написаны на языке программирования C++ в среде программирования Visual Studio 2013, для программирования GPU использовался CUDA Toolkit 7.5.

Таблица 1. Параметры используемых CPU и GPU

CPU — Intel Core i7—4790	GPU — GTX 750
Тактовая частота — 3600 ГГц Кол-во ядер/потоков — 4/8 Объем оперативной памяти — 32 ГБ Производительность — 4.18 ГФлопс/ядро ОС — Windows 7, 64 бит	Тактовая частота — 1024 МГц Кол-во ядер — 512 Объем памяти — 1 ГБ Пропускная способность памяти — 80 Гб/с

Таблица 2. Время вычисления при $N=2^h$ и $L=1000$, $v=500$, с

Метод	*h*							
	14	15	16	17	18	20	22	24
ПС на CPU	8,36	16,64	33,37	66,49	133,13	572,74	2452,69	1054,4
БС на CPU	0,172	0,384	0,760	1,687	3,487	30,118	169,854	909,174
БС на GPU	0,108	0,150	0,233	0,395	0,769	2,899	11,385	45,335
Умн. матриц на GPU	0,089	0,168	0,334	0,656				

Результаты определения скорости вычислений (усредненные по 50 реализациям) в зависимости от размера исходной выборки $N=2^h$ при использовании различных методов обработки представлены в таблице 2.

Отношение времени вычисления функции неопределенности при расчете различными методами относительно времени вычисления прямой свертки на CPU (расчет на 1 ядре) представлено на рисунке 2.

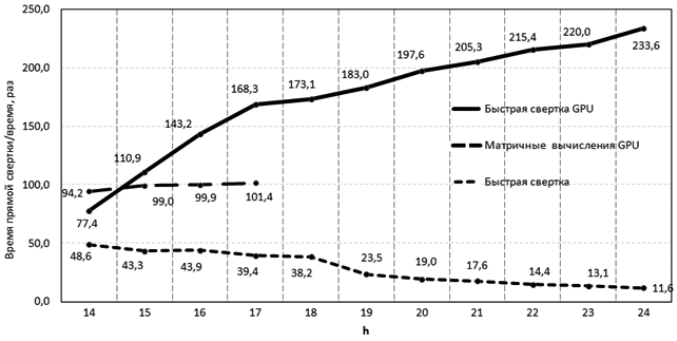

Рисунок 2. Отношение времени вычисления различными методами обработки к времени вычисления прямой свертки на CPU (1 ядро) в зависимости от объема исходных данных

Выводы

По результатам анализа, наибольшая скорость вычислений функции неопределенности при инверсном синтезировании апертуры для больших объемов входных данных обеспечивается применением метода быстрой свертки при реализации на GPU.

При малом объеме входных данных максимальная скорость обработки переменных типа float была достигнута при реализации метода матричного умножения — 604 ГФлопс, что объясняется использованием оптимизированной библиотеки cuBLAS.

ЛИТЕРАТУРА

1. *Ксендзук, А.В. Результаты обнаружения летательных аппаратов по сигналам DVB-T2 // II Московская Микроволновая неделя. — М.: ИРЭ РАН им. Котельникова, 2014. — С. 262—266.*
2. *Герасимов, П.А., Кузнецов, И.А. Аппаратно-программный комплекс для проведения экспериментальных работ и моделирования неизлучающей радиолокационной системы ПВО-ПРО // Радиопромышленность, — 2015. — № 1. — С. 182—188.*
3. *Карих, А.А. Реализация корреляционной обработки на графических процессорах // Вестник Концерна ПВО «Алмаз-Антей». — 2014. — № 2. — С. 109—114.*
4. *Шевченко, А.В., Хохлов, Н.И., Цыбулин, И.В. Исследовательские и учебные центры CUDA. МФТИ // CUDA Альманах. — 2015. — № 7. — С. 7—10. — [Электронный ресурс]. — Режим доступа: http://www.nvidia.ru/object/cuda-parallel-computing-almanac-ru.html (дата обращения 08.10.2015).*
5. *Ksendzuk A.V. Volosyuk V.K. Sologub N.S. Modeling SAR primary and secondary processing algorithms. Estimating quality of the processing techniques // 5-th European Conference on Synthetic Aperture Radar EUSAR2004. — Ulm, Germany. — Vol. 2, 2004. — p. 1013—1016.*
6. *Волосюк, В.К., Жеребятьев, Д.П., Кравченко, А.И., Ксендзук, А.В. Синтез оптимальных алгоритмов оценки параметров изображений пространственно-протяженных объектов и их местоположения // Системы вооружения и военная техника. — 2006. — № 1(5). — С. 88—94.*

7. *Солонина, А.И., Улахович, Д.А., Арбузов, С.М., Соловьева, Е.Б. Основы цифровой обработки сигналов: курс лекций. — 2-изд., испр. и перераб. — СПб.: БХВ-Петербург, 2005. — 768 с.; ил.*
8. *FFTW. Library for computing the discrete Fourier transform. — [Электронный ресурс]. — Режим доступа: http://www.fftw.org (дата обращения 8.10.2015).*

ИНФОРМАЦИЯ ОБ АВТОРАХ

Ксендзук Александр Владимирович, *д.т.н., начальник отдела, ОАО «МАК “Вымпел”», Москва, 4-я ул. 8 Марта, д.3, тел.: (499) 152-23-74, e-mail: vimpel215@mail.ru.*
Герасимов Павел Андреевич, *к.т.н., замначальника отдела, ПАО «МАК “Вымпел”», Москва, 4-я ул. 8 Марта, д.3, тел.: (499) 152-31-60 доб. 4-25, e-mail: vimpel215@mail.ru.*

For citation: Radiopromyshlennost. — 2016. — № 1. — P. 33—37.
УДК 621.396.967
A.V. Ksendzuk, P.A Gerasimov

INVERSE PASSIVE SYNTHETIC APERTURE RADAR

Paper represents digital processing optimization in passive inverse synthetic aperture radar. Processing algorithms analyzed and compared for CPU and GPU processing.

Keywords: *inverse synthetic aperture radar, passive radar, CUDA computing, DVB-T.*

REFERENCES

1. *Ksendzuk A.V. Results of detections aircraft used signal DVB-T2. IRE RAS of Kotelnikova, 2014, pp. 262—266.*
2. *Gerasimov P.A. Kuznetsov I.A Hardware-software complex for experimental works and modeling passive radar. Radiopromyshlennost, 2015, 1, pp. 182—188.*
3. *Karikh A.A. Implementation of the correlation processing on GPUs. Journal of Concern PVO «Almaz-Antey», 2014,2, pp. 109—114.*
4. *CUDA almanac. http://www.nvidia.ru/object/cuda-parallel-computing-almanac-ru.html.*
5. *Ksendzuk A.V. Volosyuk V.K. Sologub N.S. Modeling SAR primary and secondary processing algorithms. Estimating quality of the processing techniques // 5-th European Conference on Synthetic Aperture Radar EUSAR2004. — Ulm, Germany. — Vol. 2, 2004. — p. 1013—1016.*
6. *Volosyuk V.K., Zherebyatev D.P., Kravchenko A.I., Ksendzuk A.V. Synthesis of optimal estimation algorithms of image parameters spatially extended objects and their locations. Sistemy vooruzheniya i voennaya tekhnika, 2006, 1(5), pp. 88—94.*
7. *Solonina A.I., Ulakhovich D.A., Arbuzov S.M., Solovev E.B. Bases of digital processing of signals: Course of lectures. BHV-Petersburg, 2005, p.768.*
8. *FFTW. Library for computing the discrete Fourier transform. http://www.fftw.org.*