

文章编号 :1001 - 2486(2007)02 - 0061 - 04

分布式小卫星 SAR 回波仿真快速算法*

王 敏 梁甸农 董 臻 路兴强

(国防科技大学 电子科学与工程学院 湖南 长沙 410073)

摘 要 :回波信号仿真是研究分布式小卫星 SAR 系统的基础 ,对于系统总体设计具有重要意义。大范围自然场景回波仿真导致巨大的计算量。快速算法基于 FFT 实现 ,首先在时域利用脉冲序列近似表征场景在慢时刻的响应 ,然后利用 FFT 在频域实现线性时不变滤波以产生场景回波 ,算法能够有效减小运算量。对仿真的回波进行成像和干涉处理 ,结果验证了算法的有效性。

关键词 :星载 SAR ;回波仿真 ;FFT ;快速算法

中图分类号 :TN955 文献标识码 :A

A Rapid Algorithm for Distributed Small Satellite
SAR Raw Signal Simulation

WANG Min , LIANG Dian-nong , DONG Zhen , LU Xing-qiang

(College of Electronic Science and Engineering , National Univ. of Defense Technology , Changsha 410073 , China)

Abstract Raw signal simulation is considered the base for the study of distributed small satellite SAR system. Huge computation is needed when raw signal of wide area natural scenes is simulated. A rapid algorithm using FFT is proposed to reduce computation. Firstly , time domain response of the natural scene is approximately expressed by pulse sequence. Then in frequency domain , LTI filter is realized with FFT to generate raw signal. The results of SAR imaging and interferometric processing of simulated raw signal validate the algorithm.

Key words spaceborne SAR ; raw signal simulation ; FFT ; rapid algorithm

分布式小卫星 SAR 系统是近年来提出的一种新体制星载雷达系统 ,它是由编队飞行的若干颗小卫星协同完成一个任务 ,目前分布式小卫星 SAR 系统仍处于研究阶段 ,没有实际运行的系统。因此 ,回波仿真成为分布式小卫星 SAR 系统仿真的基础。大测绘带回波仿真涉及到海量运算 ,如果使用时域算法仿真地面场景 ,在当前主流的 PC 机上需要运行数月的时间 ;同时 ,随着编队飞行的小卫星数目的增加 ,运算量将线性增加 ,一般的时域仿真方法已不能满足分布式小卫星 SAR 回波仿真的要求。但文献 [1] 中的二维频率算法不适合 InSAR 回波的仿真。文献 [2] 提出了一种快速算法 ,在时域利用 sinc 函数插值来近似表征场景的响应 ,在频域利用 FFT 来减小运算量。本文提出了一种快速算法 ,首先在时域利用复脉冲序列近似表征场景的响应 ,然后将该序列输入一个线性时不变滤波器以产生场景回波 ,由于不需要计算 sinc 函数 ,计算量大幅减小 ,算法效率得到提高 ,同时通过对回波进行成像和干涉处理 ,结果表明这种近似可以满足系统仿真的要求。

1 分布式小卫星 SAR 回波仿真原理

分布式小卫星 SAR 系统有单发多收和多发多收两种工作模式。无论哪一种工作模式 ,最终的回波仿真都涉及到发射信号与接收信号的雷达之间的空间几何关系和信号模型^[3] ,相当于星载双基地回波仿真 ,当发射信号的雷达和接收信号的雷达相同时 ,和传统的单星 SAR 回波仿真相同。

设发射的基带信号为 $\mu(\tau)$,地面某散射点到发射雷达与接收雷达斜距之和随慢时间变化的函数为

* 收稿日期 2006 - 10 - 16

基金项目 : 国家部委基金资助项目(51317)

作者简介 : 王敏(1978—) ,男 , 博士生。

$R(s)$ 则正交解调后该散射点的回波信号可表示为

$$S(s, \tau) = \sigma W_{as}(s) W_{ar}(s) \cdot p\left(\tau - \frac{R(s)}{c}\right) \cdot \exp\left\{-j \frac{2\pi}{\lambda} R(s)\right\} \quad (1)$$

式中 s 为方位向的慢时间 τ 为距离向的快时间, $W_{as}(\cdot)$ 为发射天线单程方向性函数, $W_{ar}(\cdot)$ 为接收天线单程方向性函数 σ 为目标的双站雷达复散射系数, λ 为信号波长, c 为光速。分布目标(或场景)可分解为若干个散射点, 每个散射点可看成具有不同复散射系数的点目标, 场景回波可近似为这些点目标回波的叠加。若分布目标(或场景)总共分解为 N 个散射点, 回波可表示为

$$d(s, \tau) = \sum_{i=1}^N S_i(s, \tau) \quad (2)$$

2 基于 FFT 的快速回波仿真

式(2)中的 $S_i(s, \tau)$ 可写为以下等价形式:

$$S_i(s, \tau) = p(\tau) \otimes \sigma_i W_{asi}(s) W_{ari}(s) \exp\left\{-j \frac{2\pi}{\lambda} R_i(s)\right\} \delta\left(\tau - \frac{R_i(s)}{c}\right) \quad (3)$$

即 SAR 点目标距离维回波可表示成两个信号的线性卷积形式, 卷积号左边的信号 $p(\tau)$ 是雷达发射信号, 与场景中的目标散射特性无关, 卷积号右边反映了目标与雷达的相对位置等关系, 这种关系是随方位时间变化的, 可以视为场景在方位时间 s 对雷达信号的冲击响应, 则式(2)可以写为如下形式

$$d(s, \tau) = p(\tau) \otimes \sum_{i=1}^N \sigma_i W_{asi}(s) W_{ari}(s) \exp\left\{-j \frac{2\pi}{\lambda} R_i(s)\right\} \delta\left(\tau - \frac{R_i(s)}{c}\right) \quad (4)$$

令

$$h(s, \tau) = \sum_{i=1}^N \sigma_i W_{asi}(s) W_{ari}(s) \exp\left\{-j \frac{2\pi}{\lambda} R_i(s)\right\} \delta\left(\tau - \frac{R_i(s)}{c}\right) \quad (5)$$

令 $d(s, \tau) \xleftrightarrow{f} D(s, \omega)$, $p(\tau) \xleftrightarrow{f} P(\omega)$, $h(s, \tau) \xleftrightarrow{f} H(s, \omega)$, 则式(4)的距离向频域表达式为 $D(s, \omega) = P(\omega) \cdot H(s, \omega)$, 对该式反变换即可得到分布目标的回波 $d(s, \tau)$ 。

$h(s, \tau)$ 为具有一定幅度、相位、延迟的冲激脉冲串。如果直接对 $h(s, \tau)$ 进行采样, 由于 $R_i(s)$ 对应的延时 τ_i 并不总是恰好在采样点上, 因而这里采用升采样后单脉冲逼近的方法。此方法的基本思想是先对基带信号 $p(\tau)$ 进行 L 倍升采样, 对 $h(s, \tau)$ 作类似处理, 每个散射点的延迟用离其最近的采样点的延迟来代替, 对 $D(s, \omega)$ 反变换后得到的结果降 L 倍抽取, 可得到最终的回波信号。设原始的距离向采样周期为 Δt , 令 $\tau_i = \frac{R_i(t)}{c}$, $\tau'_i = \text{round}\left(\frac{\tau_i}{\Delta t/L}\right) \Delta t/L$, 则由式(4)可得

$$d'(s, \tau) = p(\tau) \otimes h'(s, \tau) \quad (6)$$

$$h'(s, \tau) = \sum_{i=1}^N \sigma_i W_{asi}(s) W_{ari}(s) \exp\left\{-j \frac{2\pi}{\lambda} R_i(s)\right\} \delta(\tau - \tau'_i) \quad (7)$$

由上式可见, 采用这种方法会在距离向包络引入一定的误差, 而相位则是精确的。

3 计算效率分析

按照“走-停-走”的模式, 如果用传统的时域叠加的方法仿真回波, 一条回波的计算量主要包括两部分。第一部分 $C_1 = NC_g$, 计算收发卫星(雷达)的位置、各散射点与收发雷达的空间角度、天线加权、 $R_i(s)$ 等。 N 为散射点的个数, C_g 为计算一个散射点空间几何关系的计算量。第二部分 $C_2 = NC_p = NT_p F_s C_c + NT_w F_s C_a$, 计算各散射点的时域回波, 并将之叠加。 C_p 为计算一个散射点时域回波的计算量, T_p 为发射信号脉冲宽度, T_w 为雷达波门宽度, F_s 为采样频率, C_c 为单个采样点复系数构成的计算量, C_a 为复数相加的计算量。则传统算法总的计算量 $C_{total1} = N_a(C_1 + C_2)$, N_a 为方位向采样点数。

按照本文中的快速算法, 计算量主要包括三个部分。第一部分 C'_1 , 同传统时域叠加的方法计算量相同, 即 $C'_1 = C_1$ 。第二部分 $C'_2 = NC_c + NC_a$, 由各散射点对应的延时计算相应的 $h'(s, \tau)$ 。第三部分 $C'_3 = 2C_{fft} + LT_w F_s C_m$, C_{fft} 为 $LT_w F_s$ 点长的 FFT 计算量, C_m 为一次复数相乘的计算量。快速算法总的计

算量为 $C_{total2} = N_a(C'_1 + C'_2 + C'_3)$ 。

对于传统方法和快速算法, 散射点与雷达的空间几何关系的计算量即 C_1 、 C'_1 计算量相同, 快速算法相对于传统算法的优势在于各散射点回波的叠加, 定义部分加速比 η 为

$$\eta = C_2/C'_2 + C'_3 = T_p F_s C_c + T_w F_s C_a / C_c + C_a + \frac{1}{N} C'_3 \quad (8)$$

式中, C'_3 与散射点的数量无关, 当升采样倍数确定后, 在仿真大型场景时, 散射点的数量很多, 因此

$$\lim_{N \rightarrow \infty} \eta = T_p F_s C_c + T_w F_s C_a / C_c + C_a \approx T_p F_s \quad (9)$$

式中, C_a 相对于 C_c 是一个较小量, 而 T_w 与 T_p 在仿真中有相同的数量级, 所以近似是可以成立的。

定义总的加速比 η_{total} 为

$$\eta_{total} = C_{total1} / C_{total2} = N_a(C_1 + C_2) / N_a(C'_1 + C'_2 + C'_3) = C_1 + C_2 / C'_1 + C'_2 + C'_3 \quad (10)$$

$$\lim_{N \rightarrow \infty} \eta_{total} \approx C_g + T_p F_s C_c + T_p F_s C_a / C_g + C_c + C_a \quad (11)$$

在 DELL 的 PowerEdge1750 (CPU 为 P4 3.2GHz, 内存为 1GB) 上测试可得到 C_g 的平均值约为 0.000386s, C_c 的平均值约为 0.000365s, C_g 和 C_c 的计算量相当, C_a 的计算量相对较小, 可以忽略, 因此

$$\lim_{N \rightarrow \infty} \eta_{total} \approx (T_p F_s C_c) / (C_g + C_c) \approx T_p F_s C_c / 2C_c = T_p F_s / 2 = \eta / 2 \quad (12)$$

文献 [2] 中算法的加速比为 $\lim_{N \rightarrow \infty} \eta' = \frac{T_p F_s}{M}$, 其中 $M \geq 8$, 为 sinc 插值的点数。

$$\eta_{total} / \eta' = M / 2 \quad (13)$$

由式 (13) 可见, 本文中的算法效率优于文献 [2] 中的算法。

设定升采样倍数为 16 倍, 以 Envisat 为主卫星, 卫星高度 759km, 信号波长 5.6275cm, 天线长 10m, 天线高 1.3m, 中心下视角 30° , 信号带宽 $B = 16\text{MHz}$ 。辅星在 Envisat 前约 100km 处, 取其中一颗接收小卫星进行分析, 纬度辐角相差主星 0.81° 。假定主星工作于正侧视模式, 设定发射信号脉宽为 $25\mu\text{s}$, 采样频率为 32MHz, 脉冲重复频率 2000Hz, 方位向采样点数为 1400, 回波距离向采样点数为 1242, 一个脉宽内的采样点数为 800。改变散射点的数量, 测试加速比得到图 1。由图 1 可见, 加速比 η 随着散射点的增加而增大, 当散射点数量较少时, 加速比远小于理论值; 当散射点数量达到 4096×4096 时, 加速比值接近理论值。总的加速比 η_{total} 也随着散射点数量的增加而增大, 当散射点数较多时, 接近部分加速比的一半。

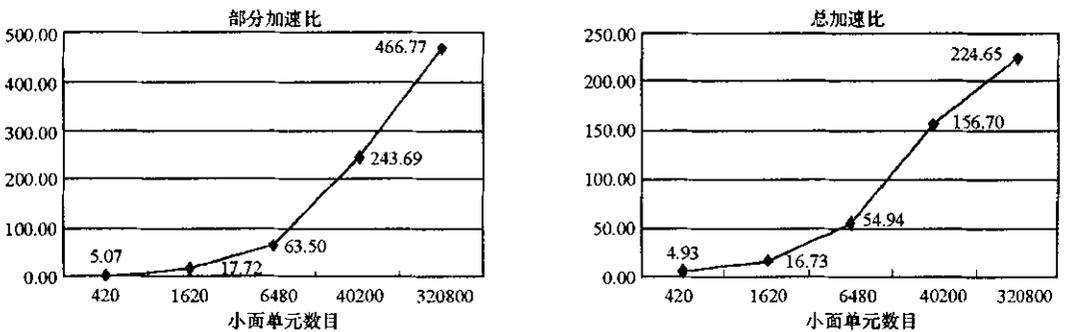


图 1 部分加速比和总加速比随散射点数量的变化规律

Fig. 1 Partial and total speedup factor characteristic varies with scatter number

4 误差分析及仿真结果

比较式 (4) 和式 (7) 可知, 采用这种方法对方位向多普勒历史不会造成影响, 但在距离向将引入随机移位, 导致成像后距离向上的目标位置误差。考虑最坏的情况, $\Delta(\tau - \tau'_i)$ 恰好处于两个采样点中间时, 距离向上时间误差 $\Delta\tau_i \leq \frac{\Delta t}{2L}$, 距离向上的位置误差 $\Delta r = \frac{\Delta\tau_i \cdot c}{2} \leq \frac{\Delta t \cdot c}{4L} = \frac{c}{4LF_s} < \frac{c}{4LB} = \frac{\rho_r}{2L}$, 其中 ρ_r 为理论斜距分辨率, 当升采样倍数足够大 ($L \geq 16$) 时, 快速算法带来的位置误差远远小于斜距分辨率。

改变升采样倍数, 比较快速算法与传统算法仿真的点目标回波, 结果如图 2 所示。当升采样倍数达

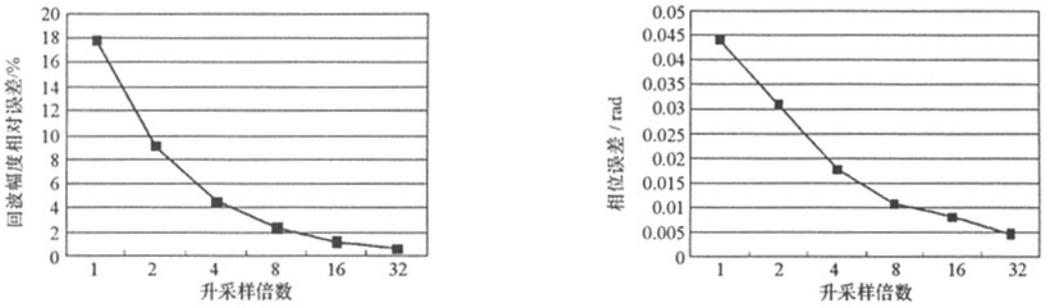


图 2 回波误差随升采样倍数的变化

Fig. 2 Echo error characteristic varies with rise-sample ratio

到 16 时,回波幅度相对误差约为 1%,回波相位误差小于 0.01rad,对成像影响不大。将升采样倍数为 16 时的点目标回波成像处理后,图像指标如表 1 所示,各项指标差别都在 0.01dB 以下,对后续干涉处理不会造成影响。当升采样倍数增加时,计算量增加,加速比会降低。对包含 4096 × 4096 个散射点的地面场景改变升采样倍数,测试加速比的变化,当升采样倍数从 16 增加为 128,加速比从 378.7 降为 335.3,仿真总时间达到 11h。为使误差在允许范围内时,仿真时间最短,在仿真时我们选择加速比为 16。

表 1 图像质量比较

Tab. 1 Image quality comparison

	方位向峰值旁瓣比 (dB)	方位向积分旁瓣比 (dB)	方位向分辨率 (m)	距离向峰值旁瓣比 (dB)	距离向积分旁瓣比 (dB)	距离向斜距分辨率 (m)
快速算法	18.10	16.63	4.9	14.15	12.91	9.4
传统算法	18.09	16.62	4.9	14.14	12.92	9.4

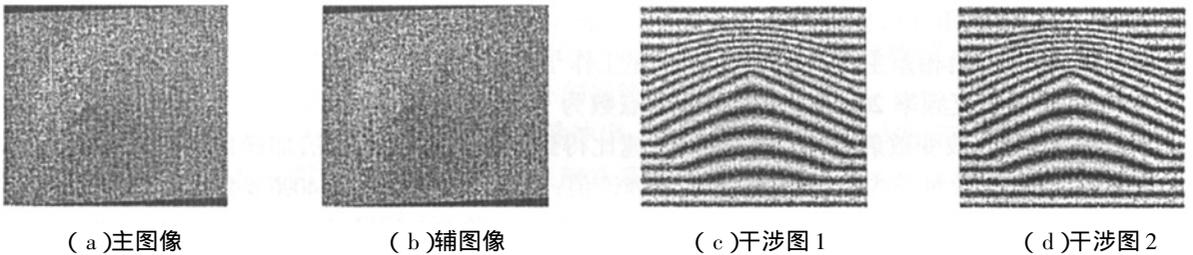


图 3 圆锥仿真结果

Fig. 3 Taper simulation results

图 3 是按文献 [4] 提供的一个圆锥场景的 RCS 数据产生的回波经过处理的结果,得到了两颗小卫星的主图像 (a)、辅图像 (b),处理产生了正确的干涉图 (c)。图 3 (d) 为传统时域算法产生的回波处理后得到的干涉图,将图 (c) 和图 (d) 比较,两者之差的均值小于 0.002rad,可见两者吻合很好,说明快速算法仿真的回波是正确的,且满足干涉处理的要求。

5 结论

由上面的分析和仿真结果可见,尽管该方法会带来回波包络上的误差,但对于 SAR 成像和 InSAR 干涉应用,基于该方法仿真的回波是可以接受的,大面积的场景和分布目标的回波仿真,如果按照传统时域方法仿真,需要的计算时间通常以月计算,对于分布式小卫星 SAR 系统仿真是不利的,快速算法大大减少了计算时间,能够在短时间内检验系统仿真的结果。

参考文献:

[1] Franceschetti G, Migliaccio M, et al. SARAS : A Synthetic Aperture Radar (SAR) Raw Signal Simulator[J]. IEEE Trans. on GRS ,1992 , 30(1) :110 - 123.
 [2] 黄立胜,王贞松,郑天垚.基于 FFT 的快速 SAR 分布目标回波模拟算法[J].遥感学报 2004 8(2) :128 - 136.
 [3] 何峰,梁甸农,刘建平.星载寄生式 SAR 系统干涉信号模型与相对高程测量性能分析[J].国防科技大学学报 2005 27(3) :72 - 76.
 [4] 孙造宇,梁甸农.星载分布式 SAR 干涉信号仿真及相干性分析[J].现代雷达 2005 27(4) :10 - 11.

