第 37 卷 第 24 期
Vol.37 No.24

计 算 机 工 程
Computer Engineering

2011 年 12 月
December 2011

• 工程应用技术与实现 • 文章编号：1000—3428(2011)24—0239—03 文献标识码：A 中图分类号：TP391

基于 Chirp Scaling 算法的相位补偿优化

石长振[1,2]，王贞松[1]

(1. 中国科学院计算技术研究所，北京 100190；2. 中国科学院研究生院，北京 100049)

摘 要：针对 Chirp Scaling 算法相位因子表达式复杂、计算量过大的问题，提出一种适合于实时成像的相位补偿因子软硬件协同设计方法。该方法结合软件的灵活性和硬件逻辑处理的快捷性，能够快速实现相位因子的生成与补偿。理论分析与测试结果表明了该方法的高效性。

关键词：Chirp Scaling 算法；软硬件协同设计；相位补偿因子；实时处理；流水线

Phase Compensation Optimization Based on Chirp Scaling Algorithm

SHI Chang-zhen[1,2], WANG Zhen-song[1]

(1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China;
2. Graduate University of Chinese Academy of Sciences, Beijing 100049, China)

【Abstract】The amount of computation is huge as a result of the Chirp Scaling(CS) algorithm's complicated phase factor expression. To solve this problem, a hardware-software co-design method which is based on phase compensation factor and suitable for real time image processing is proposed. Taking advantages of the feasibility of numerical calculation by software and the high efficiency of computation by programmable logic, it can generate and compensate the phase factor quickly and efficiently. Theory analysis and experimental results show the efficiency of the method.

【Key words】Chirp Scaling(CS) algorithm; software and hardware co-design; phase compensation factor; real-time processing; pipeline

DOI: 10.3969/j.issn.1000-3428.2011.24.080

1 概述

随着星载合成孔径雷达(Synthetic Aperture Radar, SAR)向着高分辨率、多极化、多波段和多模式方向发展，产生的回波数据远远超过卫星下传数据链路的传输能力。成像处理后的复数图像数据相关性较原始数据好，可以做高倍比的压缩以减少数据传输量。因此，实现星上实时成像处理是重要发展方向之一[1-2]。Chirp Scaling(CS)算法在宽条带、宽波束和大斜视的情况下仍能精确实现距离迁移校正和具有较好的相位保真度，因此采用 CS 算法实现星载 SAR 图像处理具有较好的前景[3-4]。但是，由于相位补偿因子的表达式较为复杂，使得因子的生成和补偿计算量较大。且 SAR 回波数据具有数据率高和数据量大的特点，对成像处理器的处理能力要求较高。而星上环境对载荷的体积、重量和功耗等的约束，成像器规模不可能很大。因此，如何快速高效实现相位补偿是采用 CS 算法设计星上实时成像器必须解决的问题。

针对 CS 算法的高效实时实现问题，国内外科研机构在这方面进行了一些研究，取得了有一定价值的成果[5-7]，但是采用这些方法设计的成像器不能同时做到处理能力强和系统规模小，无法满足星上实时处理的需求。本文提出一种适合于实时成像的相位补偿因子的软硬件协同设计方法。首先根据 CS 算法相位补偿因子计算公式的特点进行特征分析和重新推导，统一了 3 个相位因子的表达式，然后对 CS 算法的流程进行了优化设计。根据分析和优化结果，设计并实现了相位补偿因子的高效率的计算。用该方法研制的处理板卡完成 16 384×65 536 大小数据的成像处理仅为 12.5 s。

2 Chirp Scaling 算法原理及流程

经典 CS 算法[1]首先通过方位向傅里叶变换将回波信号变换到距离多普勒域(方位向是频域)，与 Chirp Scaling 因子函数 $\Phi_1(\cdot)$ 相乘，使所有距离门的距离徙动曲线补偿到相同形状；然后通过距离向傅里叶变换将信号变换到二维频域(又称波数域)，与距离补偿因子函数 $\Phi_2(\cdot)$ 相乘，完成距离徙动校正和距离压缩；再进行距离向傅里叶逆变换将信号变换回距离多普勒域，与方位补偿因子函数 $\Phi_3(\cdot)$ 相乘，完成方位处理；最后利用方位向傅里叶逆变换将信号变换回时域，得到 SAR 图像。算法的处理流程如图 1 所示。

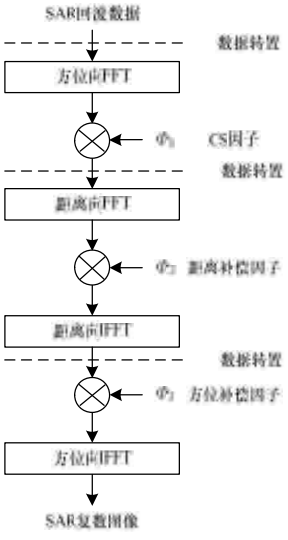

图 1 经典 CS 算法流程

基金项目：国家部委基金资助项目

作者简介：石长振(1976－)，男，博士研究生，主研方向：计算机体系结构，实时信号处理；王贞松，研究员、博士生导师

收稿日期：2011-05-16 **E-mail：**shicz@189.cn

图 1 中 3 个相位补偿因子的表达式如下：

Chirp Scaling 相位因子表达式：

$$\Phi_1(\tau,f;R_{ref})=\exp\{-\mathrm{j}\,b_r\,f\,R_{ref}\,C_s\,f\,\tau-\tau_{ref}\,f^2\} \tag{1}$$

距离向相位补偿因子表达式：

$$\Phi_2(f_\tau,f;R_{ref})=\exp\{-\mathrm{j}\,\frac{f_\tau^2}{b_r(f;R_{ref})[1+C_s(f)]}\times \exp\{\mathrm{j}\frac{4}{c}f_\tau R_{ref}C_s(f)\} \tag{2}$$

方位向相位补偿因子表达式：

$$\Phi_3(R,f)=\exp\{-\mathrm{j}\frac{4}{\lambda}R[1-\sin\phi\cdot D(f)]+(-\mathrm{j}\frac{2}{v}Rf\quad\phi)+j\Theta(f)\} \tag{3}$$

其中，因式的数学表达式和变量的物理意义如下：

$$C_s(f)=\frac{1}{\sqrt{1-(\frac{\lambda f}{2v})^2}}-1 \tag{4}$$

$$b_r(f;R_{ref})=\frac{b}{1+bR_{ref}\frac{2\lambda}{c^2}\frac{(\frac{\lambda f}{2v})^2}{[1-(\frac{\lambda f}{2v})^2]^{\frac{3}{2}}}} \tag{5}$$

$$\tau_{ref}(f)=\frac{2}{c}R_{ref}[1+C_s(f)] \tag{6}$$

$$D(f)=[1-(\frac{\lambda f}{2v})^2]^{\frac{1}{2}} \tag{7}$$

$$\Theta(f)=\frac{4}{c^2}b_r(f,R_{ref})[1+C_s(f)]C_s(f)[R-R_{ref}]^2 \tag{8}$$

$$v=\sqrt{\frac{\lambda R_{ref}f_{dr}}{2}+(\frac{\lambda f_{dc}}{2})^2} \tag{9}$$

其中，v 为雷达对地速度；R_{ref} 为参考距离；b 为发射脉冲调频率；c 为光速；λ 为雷达发射波长；f 为方位向频率；τ 为距离向时间；f_τ 为距离向频率。

3 相位补偿因子的设计优化

3.1 3 个相位因子计算表达式的简化推导和统一

式(1)~式(3)的计算非常复杂，而且计算精度要求较高，尤其是方位向补偿因子，超出单精度浮点数表示的精度范围[5]。在处理中如果按照公式直接计算，运算量非常大。如果用数字信号处理器(Digital Signal Processor, DSP)来完成因子计算，每个因子都需要上百个时钟周期。如果用现场可编程门阵列(Field Programmable Gate Array, FPGA)直接实现，需要消耗大量的片上逻辑。

从整体上来看 3 个相位因子的表达式均为方位向和距离向的二维函数，但是对于 3 个相位因子表达式中的部分因式而言却是一维函数。例如调频缩放因子 $C_s(f)$ 这个因式是方位向频率的一维函数，是 3 个相位补偿因子的表达式中都包含的因式。这个向量在一幅图像的处理过程中只要计算一次即可，而不必每次都重新计算。

这样计算量由 $O(N_r\times N_a)$量级降低到 $O(N_a)$量级，其中，N_r 和 N_a 分别为距离向和方位向的样本数。从表达式还可以看出，关于 f 的因式的表达式都较为复杂，而关于距离向尺度的表达式较为简单，3 个相位因子计算公式中都是距离向尺度变量的二次表达式，如果将关于方位向频率的部分因式的值提前计算出来，作为二次表达式的输入，就可以简化计算，降低计算量。

把 3 个相位补偿因子表达式中部分因式定义如下：

$$P_{11}(f)=\frac{2}{c}R_{ref}(1+C_s(f)) \tag{10}$$

$$P_{12}(f)=-\frac{1}{2}b_r(f;R_{ref})C_s(f) \tag{11}$$

$$P_{21}(f)=\frac{2}{c}R_{ref}C_s(f) \tag{12}$$

$$P_{22}(f)=\frac{1}{2(1+C_s(f))b_r(f:R_{ref})} \tag{13}$$

$$P_{31}(f)=-\frac{2}{\lambda}\{1-\sin\phi\cdot[1-(\frac{\lambda f}{2v})^2]^{\frac{1}{2}}\}+\frac{f\cos\phi}{v} \tag{14}$$

$$P_{32}(f)=\frac{2}{c^2}b_r(f;R_{ref})[1+C_s(f)]C_s(f) \tag{15}$$

那么 3 个相位因子的表达式可以表示为：

$$\Phi_1(\tau,f)=2\pi\left[\cdot\tau+P_{12}\,\tau-P_{11}^2\right] \tag{16}$$

$$\Phi_2(f_\tau,f)=2\pi\left[P_{21}\cdot f_\tau+P_{22}\cdot\left(f_\tau-\right)^2\right] \tag{17}$$

$$\Phi_3(R,f)=2\pi\left[P_{31}\cdot R+P_{32}\cdot\left(R-R_{ref}\right)^2\right] \tag{18}$$

将式(15)~式(17)统一为对距离向变量的二次表达式：

$$\Phi=2\pi\left[A_1\cdot X+A_2\cdot\left(X-X_{ref}\right)^2\right] \tag{19}$$

X 的取值分别为距离向时间 τ、距离向频率 f_τ 和雷达到反射点的距离 R，这 3 个变量沿距离门更新。3 个变量 τ、f_τ 和 R 都可以用线性表达式 $X=X_0+\Delta X\cdot i$ 产生，表达式中 i 为距离门的位置索引。式(19)中的参数分别取不同的值就可以计算出相应点的相位补偿因子。

式(19)中各系数和变量的取值如表 1 所示。

表 1 3 个相位补偿因子表达式参数配置

变量名称	$\Phi_1(\cdot)$	$\Phi_2(\cdot)$	$\Phi_3(\cdot)$
A_1	0	$P_{21}(\cdot)$	$P_{31}(\cdot)$
A_2	$P_{12}(\cdot)$	$P_{22}(\cdot)$	$P_{32}(\cdot)$
X_0	τ_{near}	0	R_{near}
ΔX	ΔT	ΔFs	ΔR
X_{ref}	$P_{11}(\cdot)$	0	R_{ref}

经过上述公式代换，3 个复杂相位因子计算表达式转化为一个简单的二次多项式。3 个相位补偿因子的计算转化为对 6 个因式和其他参数常量的二次多项式运算。

因式 $P_{11}(f)$、$P_{12}(f)$、$P_{21}(f)$、$P_{22}(f)$、$P_{31}(f)$ 和 $P_{32}(f)$ 为方位向频率的函数，由因式生成的数组长度为方位向处理长度 N_a。对于每幅要进行成像处理的图像数据 6 个变量只要计算一次即可。对于每条固定的距离线来说，这个 6 个变量的值为对应每个方位向频率的固定值。

3.2 CS 算法处理流程的优化设计

如图 1 所示，经典 CS 算法进行处理时，Chirp Scaling 因子补偿是在方位向 FFT 之后沿方位向进行，然后转置进行距离向处理。在距离向处理之后，转置到方位向进行方位向相位因子补偿。CS 相位因子 $\Phi_1(\cdot)$ 和方位向相位补偿因子 $\Phi_3(\cdot)$ 沿方位向进行相位补偿。

根据式(19)，由于 X 为距离向变量，在这里对 CS 算法的处理流程做一个调整。首先将数据转置到距离向之后再进行 $\Phi_1(\cdot)$ 的补偿，其次是在做完距离向逆 FFT 处理之后直接进行 $\Phi_3(\cdot)$ 的补偿，然后再转置到方位向。调整后的流程如图 2 所示，从图 2 可以看出，3 个相位因子的补偿都是沿距离向进行的。对于每条距离线来说 6 个因式是固定量，每条距离线更新一次即可。

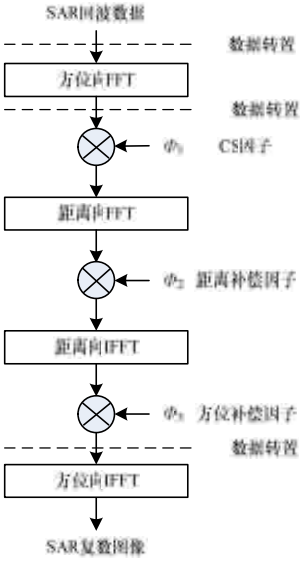

图 2　实时化的 CS 算法流程

3.3　流水线结构的相位因子的生成器的设计

根据式(19)用 FPGA 逻辑设计流水线结构的相位因子生成器，结构如图 3 所示。在处理的过程中 3 个相位因子在时间顺序上先后错开，因此可以复用一个流水线结构以节约逻辑资源。

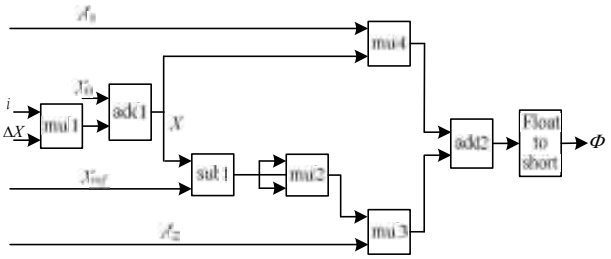

图 3　CS 相位补偿因子流水线结构

在相位补偿的过程中不同阶段，给流水线因子生成器的输入配置相应的初始化参数和变量以产生相应的因子，这种方法实现了结构可配置和资源复用，配置的参数见表 1。

4　系统的实现

4.1　信号处理板卡的结构设计

实际系统中设计了主从结构的信号处理板卡，主 FPGA 负责整个板卡的控制、处理数据的分发和收集以及各个从 FPGA 之间数据的交互。4 个从 FPGA 实现并行处理。每个从 FPGA 能够完成 CS 算法中所需要的 FFT 运算和相位因子的计算和补偿。

板卡结构如图 4 所示。

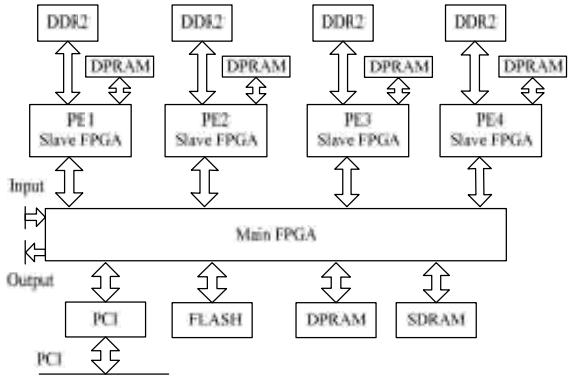

图 4　信号处理板卡结构

4.2　软件实现因式变量的计算

通过以上分析，因式 $P_{11}(f)$、$P_{12}(f)$、$P_{21}(f)$、$P_{22}(f)$、$P_{31}(f)$ 和 $P_{32}(f)$ 的计算量变为原来的 $1/Nr$。在设计中这 6 个数组计算是由主 FPGA 内部的微处理器 PowerPC405 计算完成的。由于对 CS 算法处理流程的调整，第 1 个相位因子的补偿调整到方位向 FFT 之后，为 PowerPC 计算 6 个数组赢得了时间。在构建的成像器中这部分变量的计算和方位向 FFT 的运算并行进行，且先于方位向 FFT 完成计算，PowerPC 的计算时间为方位向 FFT 的时间所掩盖，没有导致总的成像处理时间的增加。采用这种方法利用了软件进行数值计算灵活的特点，完成了用硬件逻辑不便于实现的复杂运算，且没有占用逻辑资源和处理时间。

4.3　相位补偿因子的硬件逻辑实现

由 FPGA 逻辑构建的流水结构的相位补偿因子生成器能够每个时钟周期生成一个相位补偿因子，而所耗用的逻辑资源仅为 3 个双精度浮点加(减)法器、4 个双精度乘法器和 1 个浮点至定点的转换器。图 5 为每条距离线处理的时序示意图，由于能够在每个周期生成一个相位补偿因子，因此在数据输入到 FFT 处理器或从 FFT 处理器输出的同时完成了相位补偿，没有额外耗费时间。采用这样的设计方法使得成像处理时间完全取决于 FFT 处理器的处理速度。从图 5 可以看出，相位因子的生成及补偿时间和 FFT 数据输入输出的时间完全重叠，从某中意义上可以认为生成相位补偿因子占用的时间为 0。

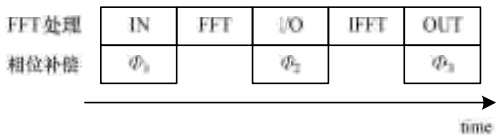

图 5　距离向 FFT 及相位因子补偿时序示意图

5　测试结果

通过实验对采用相位因子优化的实现方法和通常无优化的实现方法进行效率上的定量比较。由于优化是在算法层次进行，因此优化效率对具体的计算平台依赖性不大，这里选用 PC 机作为运算平台。处理数据的方位向长度 N_a=16 384，距离向长度 N_r=65 536。分别对 3 个相位因子的生成和补偿时间进行统计，表中时间单位为秒。表 2 为 2 种实现方法的统计结果。

表 2　优化实现和无优化计算时间对比

函数	直接计算用时/s	优化计算用时/s	提高比率
$\Phi_1(\cdot)$	852.200	214.042	3.982
$\Phi_2(\cdot)$	815.453	232.107	3.514
$\Phi_3(\cdot)$	1 153.406	224.821	5.130
$\Phi_1(\cdot)+\Phi_2(\cdot)+\Phi_3(\cdot)$	2 821.059	670.970	4.204

通过表 2 的程序仿真时间结果对比，可以看出采取优化算法效率提高为原来的 4.204 倍。

采用该优化方法设计的信号处理板进行成像测试，用实际雷达回波数据进行成像处理。图像方位向长度为 16 384 点，距离向长度为 65 536 点。FPGA 运行频率为 100 MHz，成像时间实测为 12.5 s。考虑到数据的输出时间开销，当雷达脉冲重复频率为 1 920 Hz 时，用 3 块这样的信号处理板做到 1:1 的实时率。无论是用点目标模拟数据还是用实际雷达回波数据进行测试，图像质量均满足要求，符合设计指标。

(下转第 244 页)