星载合成孔径雷达并行二维成像处理的实现 [1]

王岩亭 王贞松

(中国科学院电子学研究所 北京 100080)

摘 要 星载合成孔径雷达是一个二维成像雷达系统，其记录的雷达原始数据是二维不可分离的．所以，把雷达原始数据处理成聚焦的雷达图像是一个二维的成像处理过程．二维数字信号处理对计算系统的性能要求较高，需要很大的内存空间和很高的计算速度，传统的计算机难以胜任．本文基于大规模并行计算机平台，分析了星载合成孔径雷达的二维成像算法，设计了二维快速傅里叶变换的并行新算法，有效地实现了星载合成孔径雷达原始数据的并行成像处理．

关键词 星载合成孔径雷达，成像压缩，二维快速傅里叶变换，并行处理

中图号 TN957.7

1 引 言

星载合成孔径雷达 (SAR) 是一种主动式微波成像系统，可以全天候获取地面的高分辨率图像，此外还可以观测到地表的许多前所未知的特性，因此而备受遥感界的关注．

SAR 的分辨率包含两个方面，其一是方位向也就是沿航迹向的分辨率，其二是与之垂直的距离向的分辨率．这两个方向的分辨率都要经过对雷达原始数据的压缩处理才能获得，其中距离向采用的是传统脉冲压缩技术，方位向采用合成孔径技术 [1,2]．从信息论的角度来看，SAR 系统将地域信息弥散开，以利于噪声环境中的可靠传输，也就是说 SAR 的原始数据代表的是地域的模糊图像；要获得地域的高清晰度 SAR 图像，需要经过 SAR 成像处理系统，将原始数据压缩成聚焦图像．

SAR，尤其对于星载 SAR，其系统响应是不能被分离成距离和方位两个方向的，因此其逆过程—— SAR 的成像处理也是一个不可分离的二维系统 [3−5]．二维成像处理需要很大的内存空间，其参数的更新对处理系统的影响较大．因此使用较广泛的是距离－多普勒 (RD) 算法，但是处理中必须进行距离迁移校正，需要在距离向做时域插值运算．Alan 分析了 SAR 的二维成像处理算法的实现及其与 RD 成像处理算法之间的差异 [3]．在本文中，我们基于并行计算系统，设计并实现了一个并行的二维成像处理算法．

首先，在第 2 节我们推导了 SAR 的系统响应；然后在第 3 节，主要分析了二维的成像处理系统；第 4 节简要地回顾了常用并行计算系统的主要特点；基于并行计算系统的二维 FFT (2D FFT) 算法在第 5 节中给出；最后，在第 6 节针对 SEASAT 设计了一套完整的并行二维成像系统．

2 SAR 的系统响应

SAR 波束扫过的测绘带地域可以用二维函数来表达，如果假设雷达在接收由单个发射脉冲所产生的回波数据的期间内，雷达在航迹向没有移动，则雷达记录的原始数据可以认为是这样一个二维数据块：其中一维是方位向，雷达在运动的过程中在该方向发射脉冲信号；另一维是距离向，在该方向雷达记录脉冲信号的回波．

[1] 1998-11-06 收到，1999-03-08 定稿
国家高性能计算基金（NCIC 981009）支持，国家自然科学基金（NFNS 69896250-2）资助

雷达和测绘带目标之间的几何关系如图 1 所示. 取某一时刻雷达的位置为原点 $O(0,0)$, 以方位向和距离向为坐标轴, 建立直角坐标系, 雷达的位置用 $(x',0)$ 表示, 雷达波束的中心在方位向偏移 x_c, 该时刻波束的照射下有一点目标 $P(x,r)$.

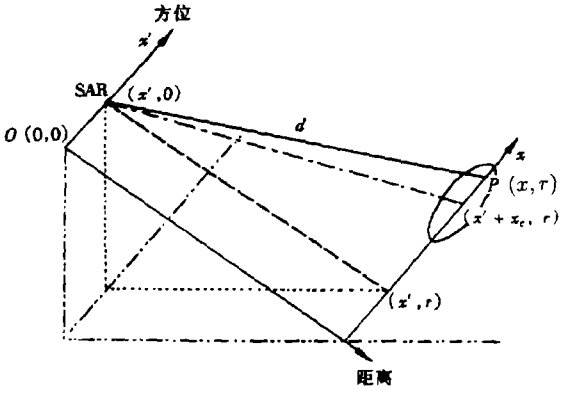

图 1 雷达平台与测绘带的几何关系

根据图 1, 该目标到雷达之间的距离为

$$d(x';x,r)=\sqrt{r^2+x'-x)^2}. \tag{1}$$

对于实际的星载 SAR 系统, 有 $r>>|x'-x|$, 所以 (1) 式可以近似写为

$$d(x';x,r)\approx r+(x'-x)^2/(2r). \tag{2}$$

记 $e(t)$ 为发射信号, 并归一化目标的散射系数为 1, 则目标 $P(x,r)$ 的回波信号为

$$r(x',t';x,r)=w(x'-x+x_c)e[t'-2d(x';x,r)/c], \tag{3}$$

这里, $w(x'-x+x_c)$ 是天线的方位向方向图所引入的加权. 记 c 为光速, 则时间变量 $t'=2r'/c$, 所以可以进一步写出

$$r(x',r';x,r)=w(x'-x+x_c)e[r'-r-(x'-x)^2/(2r)]. \tag{4}$$

由此, SAR 的系统响应即为

$$h(x',r';x,r)=w(x'+x_c)e[r'-(x'-x)^2/(2r)]. \tag{5}$$

不失一般性, 假设测绘带中心的斜距是 R_c, 我们对上式做变量代换: $r=r+R_c$, $x'=x'-x_c$. 这样, r 可以近似表示测绘带内目标离测绘带中心的位置, 加权函数 $w(x'+x_c)$ 简化为 $w(x')$, 而且对于相干系统, 幅值相对于系统的相位特性, 其重要性要小得多因而可以忽略. 为了便于分析, 我们可以从 (5) 式进一步得出一个新的系统响应表达式:

$$h(x',r';x,r)=e[r'-(x'-x_c)^2/[2(R_c+r)],\quad |x'|<L_a/2, \tag{6}$$

这里的 L_a 表示 SAR 的合成孔径长度.

接下来我们记

$$h_c(x',r') = h(x',r';0,0) = e[r' - (x' - x_c)^2/(2R_c)]. \tag{7}$$

那么

$$h(x',r';x,r) = h_c[1(x' - x_c)/\sqrt{1 + r/R_c} + x_c, r']. \tag{8}$$

假设回波信号的二维频域表示为 $H(\xi,\eta)$，根据傅里叶变换的特性可以推得

$$\begin{aligned} H(\xi,\eta;x,r) &= \sqrt{1+r/R_c}\exp\{-j\xi x_c\}\exp\{j\sqrt{1+r/R_c}\xi x_c\}H(\sqrt{1+r/R_c}\xi,\eta;0,0) \\ &= \sqrt{1+r/R_c}\exp\{-j\xi x_c\}\exp\{j\sqrt{1+r/R_c}\xi x_c\}H(\sqrt{1+r/R_c}\xi,\eta). \end{aligned} \tag{9}$$

在 SAR 系统中，发射的调制信号通常采用线性调频信号，而且其中心频率要远大于信号的带宽，所以可以用复信号来表示：

$$e(t) = \exp\{j2\pi(f_c t + k_r t^2/2)\}\mathrm{rect}(t/T), \tag{10}$$

这里，T 是脉冲宽度，f_c 是中心频率，k_r 是线性调频率．若 $x_c = 0$，根据 (7) 式和 (10) 式有

$$\begin{aligned} h'_c(x,r) = h(x,r;0,0)|_{x_c=0} &= \exp\{j4\pi r/\lambda\}\exp\left\{-j2\pi\frac{x^2}{\lambda R_c}\right\} \\ &\times\left\{j\frac{4\pi k_r}{c^2}\left[r - \frac{x^2}{2R_c}\right]^2\right\}\mathrm{rect}\left\{\left[r - \frac{x^2}{2R_c}\right]\Big/\left(\frac{cT}{2}\right)\right\}. \end{aligned} \tag{11}$$

因此，

$$H'_c(\xi,\eta) = \int \mathrm{d}x'\exp\{-j\xi x'\}\exp\left\{-j2\pi\frac{x'^2}{\lambda R_c}\right\}\int \mathrm{d}r'\exp\{-j\eta r'\}\phi(r',x'), \tag{12}$$

其中

$$\phi(r',x') = \exp\left\{j\frac{4\pi r'}{\lambda}\right\}\exp\left\{j\frac{4\pi k_r}{c^2}\left[r' - \frac{x'^2}{2R_c}\right]^2\right\}\mathrm{rect}\left\{\left[r' - \frac{x'^2}{2R_c}\right]\Big/\left(\frac{cT}{2}\right)\right\}. \tag{13}$$

根据驻定相位原理 [1]，我们可以先后求出针对 r' 和 x' 的积分，得出

$$\begin{aligned} H'_c(\xi,\eta) &= \frac{c}{2}\sqrt{\frac{1}{k_r}}\mathrm{rect}\left[\frac{c\eta/(4\pi) - f_c}{\Delta f}\right]\exp\left\{-j\left(\frac{c^2}{16\pi k_r}\eta^2 - \frac{c^2}{2\lambda k_r}\eta\right)\right\}\sqrt{\frac{2\pi R_c}{k_r\eta}}\exp\left\{j\frac{R_c}{2\eta}\xi^2\right\} \\ &= \sqrt{\frac{\pi R_c}{2k_r\eta}}c\cdot\mathrm{rect}\left[\frac{c\eta/(4\pi) - f_c}{\Delta f}\right]\exp\left\{-j\left(\frac{c^2}{16\pi k_r}\eta^2 - \frac{c^2}{2\lambda k_r}\eta - \frac{R_c}{2\eta}\xi^2\right)\right\}. \end{aligned} \tag{14}$$

对于实际的星载 SAR，上式中 ξ 的支撑集是由多普勒中心频率和多普勒带宽近似确定的．考虑相移因素，可以得到星载 SAR 在波束中心点的系统响应为

$$H_c(\xi,\eta) = \exp\{-j\xi x_c\}H'_c(\xi,\eta). \tag{15}$$

显然，从 (14) 式和 (15) 式可知 SAR 的系统响应是一个二维不可分的函数，而且从 (9) 式可以看出随着斜距的变化其系统响应是不同的.

3 SAR 的二维成像处理

因为 SAR 的成像处理是数据记录的逆过程，所以应该这样构造成像处理器

$$F(\xi,\eta;x,r)=\begin{cases}H^{-1}(\xi,\eta;x,r), & H(\xi,\eta;x,r)\neq 0;\\ 0, & H(\xi,\eta;x,r)=0.\end{cases} \tag{16}$$

因此， SAR 的成像处理也是一个不可分离的、时变的二维系统.

如果分开处理方位向和距离向，必须采取一些特殊的中间步骤对数据进行整理. 广泛使用的距离-多普勒算法，就是首先对 SAR 的原始数据进行距离向压缩，然后进行距离迁移校正；经过距离迁移校正后的数据才能接下来完成方位向压缩. 实现距离迁移校正要对大量存储数据沿距离向进行搬移，而且由于已经完成了距离向压缩处理，为了保证采集到距离压缩后信号的峰值还需要距离向插值运算. 所有这些都增大了实现的难度，降低了处理的效率，减小了聚焦的精度，而采用二维成像处理器，则不需要进行数据的插值和搬移 [3] . 然而这无疑对内存空间提出了较高的要求，需要高性能计算系统的平台支持. 此外，成像处理器在距离向是时变的，这意味着滤波器的参数要随斜距而不断地更新，也就是说随着斜距的不断变化，需要不断地重新构造成像处理器，最大的更新间隔即为距离向聚焦深度.

假设在某一时刻，目标和 SAR 之间的距离为 $R_c+r+\Delta d$ ，那么回波的相位是

$$\begin{aligned}\text{phase}&=2\pi f_c(t-2(R_c+r)/c-2\Delta d/c)+\pi k_r(t-2(R_c+r)/c-2\Delta d/c)^2\\&=2\pi f_c(t'-2\Delta d/c)+\pi k_r(t'-2\Delta d/c)^2,\end{aligned} \tag{17}$$

其中 $t'=t-2(R_c+r)/c, t\in[-T/2,T/2]$. 考虑方位向和距离向的耦合，可以求得方位向成像压缩过程中需要补偿的相位历史为

$$\varphi=2\pi 2\Delta d/\lambda-\pi k_r t4\Delta d/c+\pi k_r 4(\Delta d)^2/c^2. \tag{18}$$

相对于其他项，其中的二次项可以忽略，那么代入 $t=-T/2$ 和 Δd_m ，得上式的最大值为

$$\varphi_m=2\pi 2\Delta d_m/\lambda+\pi k_r T2\Delta d_m/c. \tag{19}$$

基于第二节的分析，我们知道有

$$\Delta d(x';x,r)=(x'-x_c-x)^2/(2(R_c+r))=[(x'-x)-2x_c(x'-x)^2+x_c^2]/[2(R_c+r)]. \tag{20}$$

所以

$$\Delta d_m=[(L_a/2)^2+|x_c|(L_a/2)+x_c^2]/[2(R_c+r)], \tag{21}$$

这里

$$L_a=\lambda(R_c+r)/(2\rho_a). \tag{22}$$

ρ_a 是 SAR 图像的方位向分辨率。

综合以上式子，就可以得到

$$\varphi_m=\pi\left[\frac{\lambda(R_c+r)}{8\rho_a^2}+\frac{|x|}{2\rho_a}+\frac{2x_c^2}{\lambda(R_c+r)}\right]+\pi\frac{\Delta f}{f_c}\left[\frac{\lambda(R_c+r)}{16\rho_a^2}+\frac{|x_c|}{4\rho_a}\right]+\frac{\pi\Delta f x_c^2}{c(R_c+r)}, \tag{23}$$

这里 Δf 是线性调频信号的带宽．进一步可以求出其对 r 的偏微分：

$$\delta\varphi_m = \delta r[\pi\lambda/(8\rho_a^2)][1 + \Delta f/(2f_c)]. \tag{24}$$

为了得到正确的聚焦，上式表达的相位误差应该被限定在 $\pm\pi/4$ 的范围内，从而就引出了距离变化间隔必须满足的条件：

$$\delta r < 2\rho_a^2/\lambda \cdot f_c/(f_c + \Delta f/2). \tag{25}$$

这里式子的右边表示的即为二维成像处理算法的距离向聚焦深度．实际应用系统中，由于有 $f_c >> \Delta f/2$，所以 (25) 式可以近似地写为 $\delta r < 2\rho_a^2/\lambda$，这也就是其他文献中推导的距离向聚焦深度 [1]．

在距离向为了满足 (25) 式，成像处理滤波器的参数应该每隔一个距离向聚焦深度更新一次．因此，处理的结果有一部分是无效的而应被抛弃，然后拼接成最终的结果，图 2 示意地描述了这一过程，其中 Nf_r 和 Nf_a 分别表示二维参考函数在距离向和方位向的长度，在星载 SAR 系统中这两个数值都很大，因此带来了二维成像算法的一个缺点：由于滤波器时变的影响，处理过程有大部分数据需要抛弃，从而处理的效率会有所降低．

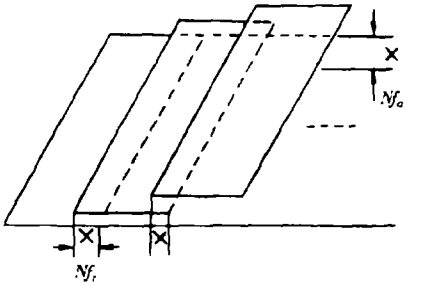

图 2　二维成像处理的数据拼接

4 并行处理系统

二维数据处理的实现需要占用很大的存储空间和花费很多的时间，所以适合选取高性能计算系统作为计算平台．伴随着通信技术和计算机技术的发展，涌现出了许多并行计算机系统，为高复杂度的问题求解提供了有力的平台支持．在众多的并行计算系统中，大规模并行处理系统 (massively parallel processing system, MPP) 和机群系统 (network of workstations, NOW) 由于较好的扩展性而得到了广泛的使用．

一个 MPP 系统由一定数量的计算节点组成，每一个节点包含一个处理器和自己独立的存储空间，节点之间通过高速网络互相连接 [6]．NOW 系统的构成和 MPP 系统相类似，但是每个节点还拥有自己的外存和 I/O 系统，具有独立计算的能力．在提交并运行一个任务之前，需要申请或指定若干个节点来使用，这些节点称为一个分区，待运行的任务就加载在所分配的分区上．由于对于运行的任务而言，其存储空间是分布的，所以在运行的过程中，各个工作节点之间可能需要进行数据交换，这是并行处理程序需要着重考虑的一个问题．一般来说，任务需要进行并行化，根据分区的大小分成几个子任务，在执行任务前把每一个子任务分别加载到各个节点上．只要任务的分解是合适的，就能使整个程序的数据处理能力成倍的增加．此外，我们知道数据的 I/O 操作一直是计算机处理速度的瓶颈所在，要比指令的执行慢得多．对于在一般计算机上的二维处理，由于数据量太大，处理过程中许多数据要暂存在外部存储空间中，这就严重影响了处理的速度．在 MPP 和 NOW 系统中，处理过程中数据可以一直保留在内存中，其交换由高速网络来完成，从而避免耗时的数据 I/O 操作．

但是需要指出的是分区中的数据交换仍然要占用一些时间，应该给予仔细的考虑．一般地，数据传输的延时包括两个方面：硬件延时和软件延时 [7]．硬件延时主要由物理的通信

网络引入，取决于通信网络的带宽，是必然存在的；而软件延时主要由系统调用和通信的初始化过程引入，与数据交换的具体设计有关.

本文中，我们主要采用的平台是曙光 1000 ——一个通用的基于消息传送机制的 MPP 系统，以及一个机群系统曙光 1000A .在表 1 中给出的是基于曙光 1000 所进行的一块 1024×1024 大小的 8 字节数据矩阵的传输延时，从中易见对于同样大小的数据块，应该尽量减少传输的次数.

表 1 不同传输次数下节点间数据传输的延时 (曙光 1000)

通信次数	1024 ×1024	1024	1
通信延时 (s)	281	7	4

总之，由于数据传输的必然存在，在我们试图提高程序的并行度时，应该考虑数据的通信量和传输次数，合理地分解任务.

5 二维 FFT 的并行实现

实现二维成像处理算法要用到二维快速傅里叶变换 (2D FFT)，与其他的二维处理一样 2D FFT 也有内在的并行性，适合于并行运算. 我们应该做的是考虑数据传输延时，对任务进行仔细分析，根据任务的大小和分区的大小，设计任务的分解方法 [8−12] .

首先我们回顾传统的 2D FFT 及其并行实现. 二维离散信号 $x(n,m)$, $n \in [0,N), m \in [0,M)$ 的 2D DFT 定义为

$$X(k,l) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} x(n,m) W_N^{nk} W_M^{lm}, \quad k = 0,1,\cdots,N-1; \quad l = 0,1,\cdots,M-1. \tag{26}$$

显然这个转换可以通过两个步骤来执行，分两个方向 (行向或列向) 先后运算:

$$y(n,l) = \sum_{m=0}^{M-1} x(n,m) W_M^{lm}, \quad l = 0,1,\cdots,M-1, \tag{27}$$

$$X(k,l) = \sum_{n=0}^{N-1} y(n,l) W_N^{nk}, \quad k = 0,1,\cdots,N-1. \tag{28}$$

这就是行列算法，其优点是可以利用现有的一维 FFT 模块. 沿任一方向，各个一维 FFT 之间是相互无关的，可以在并行平台上同时执行；而在两个步骤之间，存在转角变换问题. 从表 1 中我们可以知道，为了保证数据传输的效率，每一次传输的数据量相对于总的传输次数应该足够大，以减小传输的软件延时. 尽管根据数据当前所在的节点和位置可以判断出该数据转角后的目标节点和位置，转角变换很易实现，但是逐个数据的传输将使软件延时十分显著；此外在数据的交换过程中为了避免通信堵塞，频繁的数据传输要引入许多同步操作，这进一步增加了数据交换的时间，最终降低算法的性能.

另一种常用的算法是向量基算法，把信号 $x(n,m)$ 按数据块在 “时域” 或 “频域” 逐步选取，最终减小运算量 [13] . 以时域选取的向量基算法为例，假设 r_1 和 r_2 分别为 N 和 M 的分解因子且有 $P = N/r_1$, $Q = M/r_2$ ，则可以如下分解信号:

$$\begin{aligned} n &= r_1 p + u, \quad p = 0,1,\cdots,P-1; u = 0,1,\cdots,r_1-1; \\ m &= r_2 q + v, \quad q = 0,1,\cdots,Q-1; v = 0,1,\cdots,r_2-1. \end{aligned} \tag{29}$$

同时考虑变量代换

$$\begin{aligned} &k = i + jP, \quad i = 0,1,\cdots,P-1; j = 0,1,\cdots,r_1 - 1, \\ &l = s + tQ, \quad s = 0,1,\cdots,Q-1; t = 0,1,\cdots,r_2 - 1. \end{aligned} \tag{30}$$

可以最终推得

$$\begin{aligned} X(i+jP, s+tQ) &= \sum_{u=0}^{r_1-1}\sum_{v=0}^{r_2-1} X_{PQ}(i,s;u,v) W_N^{(i+jP)u} W_M^{(s+tQ)v} \\ &= \sum_{u=0}^{r_1-1}\sum_{v=0}^{r_2-1} X_{PQ}(i,s;u,v) W_N^{iu} W_M^{sv} W_{r_1}^{ju} W_{r_2}^{tv}. \end{aligned} \tag{31}$$

其中

$$X_{PQ}(k,l;u,v) = \sum_{p=0}^{P-1}\sum_{q=0}^{Q-1} x(r_1 p + u, r_2 q + v) W_P^{kp} W_Q^{lq},$$

$$k = 0,1,\cdots,N-1;\ \ l = 0,1,\cdots,M-1;\ \ u = 0,1,\cdots,r_1-1;\ \ v = 0,1,\cdots,r_2-1, \tag{32}$$

按照这种方法持续分解，就可以减少运算量，并且消除了转角变换．但是，向量基算法是针对数据子块操作的，这使得任务分解和选择节点个数比较困难．比如，r_1 和 r_2 通常选为 2，那么就应有 N 等于 M，而且要求并行处理时节点数为 4 的整数次幂．

由上分析可见在并行处理时，行列算法和向量基算法各有利弊．本质上，2D DFT 是数据的块处理，向量基算法就是针对数据块，利用数据块之间的相关性，减少数据交换的次数．我们可以在数据传输效率、算法的灵活性之间做一折衷：把向量基算法退化为在其中一个方向上的块操作，那么就可以在保证数据传输效率的前提下提高算法的灵活性．根据节点数，沿数据的某一个方向即行或列，连续地将数据分解并加载到各个计算节点上，对于每一个节点，可以采用任何一种传统的 2D FFT 算法；然后把整个节点上的数据作为接下来 1D FFT 抽取的一个样本，叠算出最终的结果，因此后面我们称这种算法为一维块处理算法．如果 1D FFT 抽取是基 2 的，那么分区的大小就限定为 2 的整数次幂，而不是 4 的整数次幂．

数据的输入应该给以正确的逆序，这可以在数据的分解时完成．我们知道，基 2 FFT 中，输入数据逆序是因为时域数据的持续奇偶抽取造成的[14]，图 3 描述了基 2 的情形．如果将输入数据分解 x 次，那么只有数据序号的低 x 二进制位需要逆转．此外，我们还要计算旋转因子．在一维块处理的 1D FFT 阶段，旋转因子的计算应该和全长的传统 FFT 一致．实现了这两点后，这个算法模块就具有较大的通用性，适合于通用目的的并行处理系统．

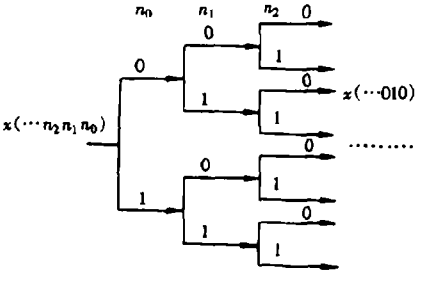

图 3　基 2-FFT 的输入逆序

6 二维 SEASAT 成像处理

最后我们以星载 SAR SEASAT 为例，来实现星载 SAR 的并行二维成像算法．处理的原始数据是伦敦地区的一块雷达数据，在方位向和距离向分别是 8192 点和 4096 点．SEASAT 的参数如表 2 所示．

表 2 SEASAT 的有关参数

脉冲宽度: 33.9μs	调频宽度: 19.0MHz
采样频率: 45.53MHz	中频频率: 11.38MHz
多普勒中心频率: −940Hz	多普勒调频斜率: −504Hz/s

方位向的处理效率可以靠增大原始数据在方位向的处理长度来提高. 但是, 在距离向的处理长度由 (25) 式表达的距离向聚焦深度所限制, 因此距离向的处理效率的提高需要尽可能地减小参考函数的长度 Nf_r 。在二维处理中, 这可以通过对原始数据在距离向的预压缩完成。

距离向的预压缩可以是完全的或是不完全的, 与 RD 算法一样, 这两者都重新引入了数据插值问题. 在接下来的二维处理中, 我们必须解决数据插值的问题。在二维处理中, 无疑将利用 2D FFT 在频域处理, 我们知道信号的频域表示即其频谱是连续的, 而且信号时域与频域的关系根据傅里叶变换对是一一对应的。假设一个信号 $f(t-\tau)$, 其对应的频谱为 $F(j\omega)\exp(-j\omega\tau)$, 将信号在时域以间隔 Δ 进行采样得到离散信号 $f(i-\tau/\Delta)$, 现在我们希望得到这个离散信号的频谱 $F(k)$ 。根据 DFT 的定义, 我们知道 DFT 变换对对应于连续信号的周期延拓的时域采样及其频谱的主周期, 所以如果离散信号的样点数为 N , 那么在频域样本之间的间隔为 $1/(N\Delta)$, 离散信号的频谱即为

$$F(k)=F[j2\pi k/(N\Delta)]\exp\{-j2\pi k\tau/(N\Delta)\}. \tag{33}$$

对于这里存在的插值问题, 实际上就是因为信号的最大能量点 τ/Δ 在离散的时域可能是不可表示的而必须通过相邻点来确定它。从上式中显然可见如果直接在频域来表达信号的话, 可以由实数的频谱值而得以完整地表达。因此, 通过频域的表示和处理可以完全消除插值问题。

首先进行距离向预处理, 因为接下去将消除插值问题, 这里采取距离向完全预压缩的方法, 同时完成 I/Q 正交解调。距离向预压缩的参考函数的长度为

$$N_{fr}=T\cdot f_s\approx 1543.$$

考虑 I/Q 正交解调, 距离向预压缩后在距离向的有效数据个数为 $(4096-N_{fr})/2\approx 1276$, 处理后的数据在距离向的有效个数为 1276 , 我们保留其中的 1024 个数据。

方位向参考函数的长度为

$$N_{fa}=T_s\cdot Prf\approx 4096,$$

这里 T_s 代表合成孔径时间。如果成像结果进行 4 视处理, 那么方位向最终的有效数据个数为 $(8192-N_{fa})/4=1024$ 。因此成像的结果为一块 1024×1024 的雷达图像.

此外, 如果我们希望处理出分辨率为 25×25m 的 SAR 图像, 即有 $\rho_a=25$ m , 根据表 2 中的数据和 (32) 式可以算出距离向的聚焦深度约为 5319m , 考虑到 I/Q 正交解调的因素, 聚焦深度反映在数据上就是 $N_r=(5319/c)f_s\approx 807$ 。也就是说, 成像时在距离向的最大处理长度为 807 点。

因此接下来要处理的数据块大小变为 8192×807 . SEASAT 的斜距能够由多普勒中心频率 f_{dc} 和多普勒调频率 k_a 表示, 类似第 2 节的推导, 我们可以得到接下来的二维处理参考函数 [1,2]:

$$h(s,t)=\exp\{j2\pi[f_{dc}s+k_as^2/2]\}\mathrm{rect}\left(\frac{s}{T_s}\right)\delta\left\{t-\frac{-\lambda f_{dc}s-\lambda k_as^2/2}{c}\right\}, \tag{34}$$

这里 s 和 t 分别代表方位向和距离向的时间变量，λ 代表雷达的波长，c 代表光速。聚焦的雷达图像就可以由预压缩后的数据和参考函数 $h(s,t)$ 的二维相关得到，而二维相关运算由并行的 2D FFT 来实现。

本文选用的分区大小为 8 个计算节点，将雷达数据沿方位向分为 8 个子数据块，2D FFT 采用一维块处理算法，那么在二维处理时每一个子块的大小为 1024×807，因参数更新引入的数据拼接依据图 2 所示方法。

数据的逆序在原始雷达数据读入时完成，以减少数据交换的需要，逆序的方法为

$$i\text{-th row in } id \text{ Node} : 8 \cdot i + \text{inverse}(id \text{ Node}) \quad 0 \le i < 1024,\ 0 \le id \text{ Node} < K$$

每一个节点上的 2D FFT 用行列算法来实现。经过全部处理后，重构得的 SEASAT 雷达图像如图 4 所示。这是一个 1024×1024 点的 4 视 SAR 图像，四视处理的目的是降低图像上的相干斑噪声。在曙光 1000 上，整个的处理时间约为 350s，而在曙光 1000A 上进一步减少到约 $150 \sim 160$s。

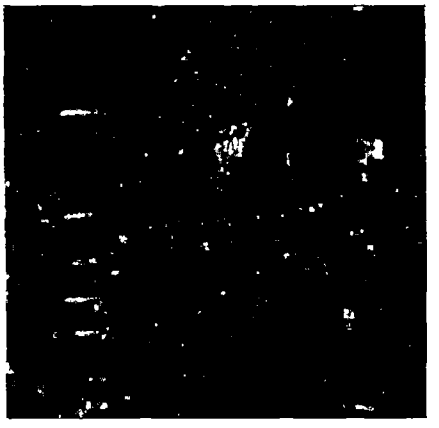

图 4　SEASAT 雷达图像 (1024×1024)

7 小　结

分析表明，SAR 的成像处理本质上是一个二维过程，而且在距离向是时变的。二维处理需要高性能的计算平台，而且适合于并行处理。但是成像处理的时变特性将会限制二维处理的总体效率。为此，首先进行距离向的预压缩是可取的，由此引入的插值问题可以通过频域运算而解决。在处理的并行实现时，要着重考虑数据通信带来的延时问题，本文基于 MPP 和 NOW 的特点，设计和实现了二维 FFT 的一维块处理算法，并利用这一算法成功地完成了 SEASAT 的二维成像处理。

参　考　文　献

[1] Curlander J C, McDough R N. Synthetic Aperture Radar. Systems and Signal Processing, New York: Wiley, 1991, Chapter, 3–4.

[2] Fitch J P. Systhetic Aperture Radar. New York: Springer-Verlag, Chapter 1, 1988.

[3] Cenzo A D. A new look at nonseparable synthetic aperture radar processing. IEEE Trans. on AES, 1988, 24(3): 218–223.

[4] Franceschetti G, Schirinzi G. A SAR processor based on two-dimensional FFT codes. IEEE Trans. on AES, 1990, 26(2): 356–365.

[5] Franceschetti G, Lanari R. Efficient and high precision space-variant processing of SAR data. IEEE Trans. on AES, 1995, 31(1): 227–237.

[6] 孙凝晖，等. 曙光 1000 大规模并行计算机系统软件的设计. 计算机学报，1997, 20(3): 259–268.

[7] 吴礼发，等. NOW 环境下并行计算中的通信时延问题. 计算机科学，1997, 20(3): 259–268.

[8] Bergland, G D. Fast Fourier transform hardware implementations— An overview. IEEE Trans. on Audio Electroacoust, 1969, 17:(6) 104–108.

[9] Corinthios M J. The design of a class of fast Fourier transform computers. IEEE Trans. on Comput., 1971, 20(6): 617–623.

[10] Gottlieb, P, *et al.* Parallel data streams and serial arithmetic for fast Fourier transform processors. IEEE Trans. on Acoust. Speech Signal Process., 1974, ASSP-22(4): 111–117.

[11] Bergland G D. A parallel implementation of the fast Fourier transform algorithm. IEEE Trans. Comput., 1972, 21(4): 366–370.

[12] Jamieson L H, Mueller P T, *et al.* FFT Algorithms for SIMD parallel processing system. J. Parallel and Distributed Computer, 1986, 3(1): 48–71.

[13] 蒋增荣，曾永泓，余品能. 快速算法. 长沙：国防科技大学出版社，1993 年.

[14] Oppenheim A V. Digital Signal Processing. Englewood Cliffs, N.J.: Prentice-Hall, 1975, Chapter 6.

THE PARALLEL IMAGE PROCESSING OF THE SPACE-BORNE SAR DATA

Wang Yanting Wang Zhensong

(*Institute of Ecletronics, Academia of Sciences, Beijing* 100080)

Abstract Principally the raw data of the space-borne synthetic aperture radar is two-dimensional Two- dimensional filters must be used to fulfill the transformation from the raw radar data to the focused image. A large memory space will be needed in the two-dimensional processing, and much computing time will be consumed. As a result, its implementation is very tedious in traditional computers. In this paper, the two-dimensional imaging processing is analyzed in detail based on parallel processing systems and the result shows that the processing time is short.

Key words Space-borne Synthetic Aperture Radar (SAR), Imaging processing, Two-dimension FFT, Parallel processing

王岩亭： 男，1974 年生，硕士，现在中国邮电工业总公司中讯通信发展有限公司工作.
王贞松： 男，1945 年生，教授，博士生导师，从事无线电物理，微波遥感研究工作.