

# 合成孔径雷达 快速后向投影算法

Ali F. Yegulalp

**摘要**——本文提出一种在时域上处理合成孔径雷达数据的新后向投影算法。该算法在任何误差要求下重现由标准后向投影算法生成的逐像素图像,但对于  $N \times N$  像素图像,它的运行速度大约要快  $\sqrt{N}$  倍。快速后向投影保留了标准后向投影的优点:对于任何飞行路径的完美运动补偿,低伪影水平,无限制的场景大小,对于任意带宽和相干积累角的完美聚焦,和严格意义上局部处理 (local processing)(即脉冲可以在采集时处理,而不需要沿着轨迹缓存 (buffering) 或转角 (corner turn)<sup>1</sup>)。新算法还可以将正在处理的图像存储在磁盘上(而不是内存中),而这对处理速度仅有轻微的影响。

**关键词**——合成孔径雷达 (synthetic aperture radar, SAR), 后向投影 (backprojection), 卷积后向投影 (convolution backprojection, CBP), 叶簇穿透 (foliage-penetrating, FOPEN) 雷达, 超宽带雷达 (ultra-wideband radar, UWB), 成像 (image formation), 聚焦 (focusing)。

## I. 引言

SAR 的成像算法有两大类: 基于 FFT 的方法和后向投影 (时域) 方法。这两类方法都有其相对优势, 而且目前都在使用。

基于 FFT 的方法倾向于最小化处理负荷, 但它们并非普遍适用的。距离-多普勒 (range-Doppler) 和极坐标格式 (polar formatting) 算法 [1], [2], [3] 依赖于几何近似, 在各种限制下 (大的图像尺寸、宽的带宽或相干积累角等) 会出现问题。更为复杂的  $\omega$ - $k$  算法 [4], [5], [6], [7], [8], [3] 在几何上是精确的, 但它仍然假定飞机匀速直线飞行。偏离轨道的运动误差只能以近似的方式进行补偿, 这可能会导致大相干积累角和/或大的运动误差的聚焦效果不佳。飞机沿轨迹的速度变化要求以均匀的形式对

数据重采样。大多数基于 FFT 的方法的另一个问题是, 它们需要对频域中的数据进行插值; 插值误差会导致扩散到整幅图像上的伪影。线性调频变标 (chirp scaling) 算法 [9], [10], [11] 不需要频域插值, 但它依赖于 Stolt 变换的一种近似, 而这种近似在低频的高分辨率图像中不成立 [3]。这些方法的困难也与算法的实现有关, 特别是对于实时处理的场景。为了计算沿轨迹的 FFT, 在开始处理之前, 必须获得一个完整孔径的脉冲回波并进行缓存。二维 FFT 需要转角操作, 这可能会成为并行计算架构的一个处理瓶颈。

通过使用后向投影, 前面的每个问题都被解决了, 代价是处理时间大大增加。本文描述了一种称为“快速后向投影”新算法, 它保持了标准后向投影的结果和优点, 同时运算速度更快。该算法是在 DAPRA 赞助的麻省理工学院林肯实验室的叶簇穿透 (FOPEN) 项目过程中开发、实现和使用的。在处理 CARABAS II VHF SAR [12] 的数据时, 我们通常获得大约 20 倍的速度提升。

本文其余部分的组织如下: 我们首先回顾标准后向投影。回顾也有助于我们建立基本的假设和符号。我们将描述一种标准后向投影的合理实现, 并分析其运算量。然后论文转向快速后向投影的讨论。在简要介绍快速后向投影的工作原理之后, 我们介绍了该方法基于基本采样原理的一个简单的图像演示。在这个演示之后, 还对采样要求进行了严格的分析, 并对如何实现该算法进行了描述。然后分析了快速后向投影的运算量, 并与标准后向投影的进行比较。本文最后使用合成的数据对标准后向投影和快速后向投影进行了对比。

## II. 标准后向投影

### A. 数据模型

为了进行后向投影, 我们需要雷达的运动历程和一组距离压缩 (range-compressed) 脉冲。用向量  $\vec{q}(s)$  表示雷达的位置, 通过路径长度  $s$  参数化<sup>2</sup>。在位置  $\vec{q}(s)$ , 我们接收到距离压缩数据  $F_0(s, t)$ 。定义时间变量  $t$  在  $t = 0$  时对应距离为零的回波。我们做了一个标准的近似, 即在发射和接受脉冲时, 雷达不会移动很大的位移<sup>3</sup>。

举一个例子, 我们考虑一个位于位置  $\vec{a}$  的理想点目标的情况。假设雷达工作在频率  $\nu_{\min}$  和  $\nu_{\max}$  之间, 并且数据在频谱中有平缓的幅度和线性相位。在时域中, 数据由

$$F_0(s, t) = \frac{1}{R^2} e^{j2\pi\vec{v}(t-2R/c)} \text{sinc}[B(t-2R/c)], \quad (1)$$

Ali F. Yegulalp 是麻省理工学院林肯实验室 (MIT Lincoln Laboratory) 的技术人员。邮箱: yegulalp@ll.mit.edu.

<sup>1</sup> “转角”是对数据对象副本存储顺序的一种变换 (例如转置) [13]——译者注, 后文中同样颜色的图文均为译者注记, 不再重复说明。

<sup>2</sup> 定义  $s$  满足  $|\mathrm{d}\vec{q}(s)/\mathrm{d}s| = 1$ 。

<sup>3</sup> “准静止”假设 [16] 或“走-停 (start-stop)”假设。

给定, 其中  $R = |\vec{q}(s) - \vec{a}|$  = 目标距离,  $B = \nu_{\max} - \nu_{\min}$  = 带宽, 以及  $\bar{\nu} = \frac{1}{2}(\nu_{\min} + \nu_{\max})$  = 中心频率. sinc 函数定义为  $\text{sinc}(t) = \frac{\sin(\pi t)}{\pi t}$ . 注意我们所定义的  $F_0(s, t)$  中包含载波相位, 并且它也包含自由空间传播的衰减因子  $1/R^2$ .

## B. 后向投影方程

后向投影的第一步是用斜坡滤波器 (ramp filter)<sup>4</sup> 对时域数据  $F_0(s, t)$  滤波. 在频谱中, 滤波可以表示为

$$\tilde{F}(s, \nu) = \tilde{F}_0(s, \nu)|\nu|. \quad (2)$$

函数  $\tilde{F}(s, \nu)$  和  $\tilde{F}_0$  分别是  $F(s, t)$  和  $F_0(s, t)$  对  $t$  的傅里叶 (Fourier) 变换.

位于位置  $\vec{p}$  的像素值是通过滤波后的时域数据进行积分得到的:

$$I(\vec{p}) = \int F\left(s, \frac{2}{c}|\vec{p} - \vec{q}(s)|\right) ds. \quad (3)$$

在实际应用中, 我们的数据是离散采样的, 所以必须用求和来代替积分:

$$I(\vec{p}) = \sum_k F\left(s_k, \frac{2}{c}|\vec{p} - \vec{q}(s_k)|\right) \Delta s_k. \quad (4)$$

注意, 离散求和式中的孔径位置 (aperture point)  $s_k$  不需要以规则的方式间隔排列. 只要孔径采样率高于 Nyquist 采样率, 离散求和就近似于积分. 连续积分式和离散求和式之间的对应关系是显然的, 所以我们在下文中仅用连续形式表示.

后向投影方程适用于任何形状的轨迹, 在原理上它可以得到一个三维图像. 对于像圆这样的强非线性轨迹, 实际上可以在空间中的三个方向都得到一定的分辨率. 对于直线轨迹, 三维成像函数  $I(\vec{p})$  沿着环绕飞行轨迹的圆圈上的值是完全恒定的. 在这种情况下, 人们通常将成像点限制在一个单一的水平面上, 我们称之为“成像平面”. 如果地面地形是已知的, 我们也可以计算出真实地面上的点的像素值  $I(\vec{p})$ . 虽然直线情况是最简单的, 但本文提出的算法基本上可以应用于任何类型的飞行路径.

## C. 积分孔径 (Integration Aperture)

在应用式 (3) 时, 必须对积分孔径 (即  $s$  的积分限) 做一些明确的选择. 最简单的选择是每个像素都使用整个可用的孔径, 这通常被称为“聚束 (spotlight) 模式”. 这种方法可以得到分辨率尽可能高的图像, 尽管点响应函数 (point response function) (和分辨率) 在空间上并不一致.

<sup>4</sup> 这是为了控制低频区域周围傅立叶变换图像的重叠 [14].

另一种常见的选择是条带 (stripmap) 处理模式, 它对每个像素使用一个固定的相干积累角. 这个角度是指飞机航线与像素的视线 (line of sight) 之间的角度. 如果我们想要从角  $\theta_1$  积累到角  $\theta_2$ , 式 (3) 中对  $s$  的积分限将由下式决定:

$$\cos(\theta_2) \leq \frac{[\vec{p} - \vec{q}(s)] \cdot \vec{q}'(s)}{|\vec{p} - \vec{q}(s)|} \leq \cos(\theta_1). \quad (5)$$

注意, 导数  $\vec{q}'(s)$  就是与轨迹  $\vec{q}(s)$  相切的单位向量<sup>5</sup>. 事实上,  $\vec{q}'(s)$  是一个单位向量是因为我们将  $s$  定义为沿轨迹的路径长度.

由于聚束与条带模式之分对这里所展现的结果并非关键, 我们将把讨论范围限制在聚束模式的情况下, 且需清楚所有内容在条带模式中也是同样直观的.

## D. 实现

在讨论后向投影时, 我们一直考虑以逐像素 (pixel-by-pixel) 的方式计算. 换言之, 我们选择一个像素, 在孔径上积分计算出它的值, 然后转到下一个像素. 在实践中, 逐脉冲 (pulse-by-pulse) 的工作方式更有意义. 对每个孔径位置  $s$ , 我们有一个对欲得图像上的每个像素都有贡献的脉冲数据  $F(s, t)$ . 换句话说, 我们是并行地计算像素值, 而不是串行的. 逐脉冲工作的优点是我们只需要访问每个脉冲一次, 而当涉及到插值时也会有好处 (后文将会解释).

式 (2) 中的滤波处理可以使用时域 FIR 滤波器或简单地用 FFT 来实现. 式 (3) 中的积分要求我们对  $F(s, t)$  关于  $t$  进行积分. 对  $F(s, t)$  关于  $t$  进行上采样 (upsample)<sup>6</sup> 是一种合理的方法. 然后, 积分值可以通过上采样后的  $F(s, t)$  的线性或最近邻 (nearest-neighbor) 插值估计得到. 可以跳过上采样步骤, 但在 Nyquist 采样率附近采样得到的数据线性或最近邻插值后会产生差的结果. 此外, 上采样步骤实际上带来的计算负担并不大, 因为它是在一个一维的数据向量上执行的, 而积分是针对二维图像中的每个像素. 经验表明, 我们可以在几乎不影响速度的情况下对数据进行六倍上采样. 后向投影的速度太慢以至于很难注意到上采样带来的额外负担这样的观点有失偏颇.

最后, 应该注意在上采样和插值之前, 必须先去除  $F(s, t)$  的载波相位, 然后在后向投影的时候再恢复相位.

<sup>5</sup>  $\vec{q}'(s)$  就是飞机当前位置在航迹上的切线方向上的单位向量.

<sup>6</sup> 所谓“上采样”, 我们具体指的是对数据间隔插零, 然后做低通滤波; 有些人可能更喜欢称之为带限插值 (band-limited interpolation). 上采样在 [15] 中有详细的阐述, 也可以在频域补零实现上采样.

## E. 计算负荷

假设使用标准后向投影, 用  $N_{pulse}$  个脉冲的距离压缩数据形成一幅  $N \times N$  像素的图像. 如果我们采用聚束模式处理, 式 (3) 在整个可获得的孔径上积分. 运算量规模为

$$N_{op} \approx N^2 N_{pulse}. \quad (6)$$

我们忽略了滤波和上采样所需的运算, 因为它们只是对式 (6) 做一些影响不大的修正.

## III. 快速后向投影

快速后向投影的工作原理是将整个合成孔径分割成子孔径. 每个子孔径生成一幅图像, 最终的图像是由各子孔径图像相干相加 (coherently summing) 得到. 关键在于, 子孔径成像在横向距离 (cross range) 上的分辨率非常低; 这意味着我们可以在横向上一个粗糙 (coarse)<sup>7</sup> 的像素网格上计算像素值. 当要将子孔径成像合并为最终的高分辨率图像时, 我们将子孔径图像在横向距离上采样到全分辨率 (full resolution). 由于子孔径成像横向距离所包含的像素远少于最终图像的, 我们需要执行的运算量远远少于标准后向投影.

在接下来的内容中, 我们将解释对子孔径使用局部坐标系的必要性、推导出采样要求、描述一个合理的算法实现, 并估计总的运算量.

### A. 子孔径处理

假设我们有一个总长度为  $L$  的 (任意形状的) 合成孔径. 后向投影方程为

$$I(\vec{p}) = \int_0^L F\left(s, \frac{2}{c} |\vec{p} - \vec{q}(s)|\right) ds. \quad (7)$$

我们可以将完整的孔径分解为  $N_{sub}$  个长度  $l = L / N_{sub}$  的子孔径. 图像  $I(\vec{p})$  变成子孔径图像  $I_n(\vec{p})$  的和:

$$I(\vec{p}) = \sum_{n=1}^{N_{sub}} I_n(\vec{p}), \quad (8)$$

其中子孔径图像由

$$I_n(\vec{p}) = \int_{-l/2}^{l/2} F\left(s_n + \xi, \frac{2}{c} |\vec{p} - \vec{q}(s_n + \xi)|\right) d\xi. \quad (9)$$

给定. 第  $n$  个子孔径的中心为  $s_n = \left(n - \frac{1}{2}\right)l$ .

式 (8) 没有做任何近似. 我们只是把 (7) 中的积分拆分成一系列的子积分 (subintegral). 从这里开始, 我们将专注于高效地计算单个子孔径的成像.

<sup>7</sup> 这里的“粗糙”意为网格单元距离大, 成像分辨率低.

## B. 笛卡尔与极坐标网格

图 1 是使用一个短子孔径生成的一幅后向投影图像. 子孔径的中心位于图像上方的某个地方. 注意, 我们在距离向 (range direction) 有很高的分辨率, 但在横向上的分辨率很低. 成像区域对应一个大的角度, 所以恒定距离的线条很明显是圆弧, 而非直线. 图 2 显示了图 1 的二维 FFT. FFT 数据填充了大部分的窗口说明图像是在 Nyquist 采样率附近采样的, 尽管横向上的分辨率很差. 如果我们在图 1 中使用较大的像素间距, 就会造成混叠 (aliasing).

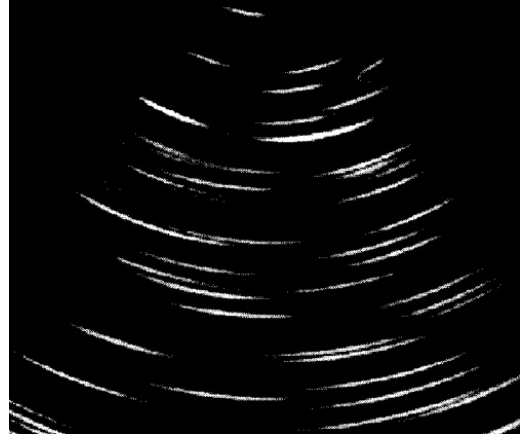


图 1 短子孔径的成像

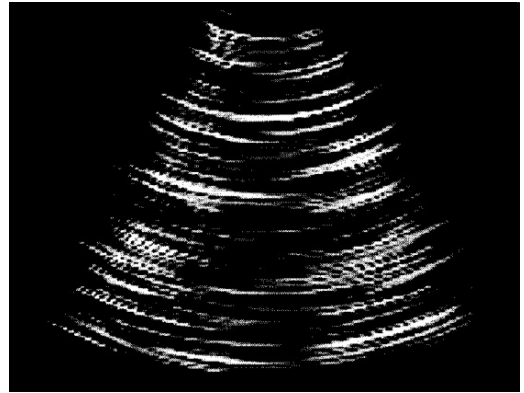


图 2 图 1 的二维 FFT

为了利用横向距离的低分辨率, 我们必须利用分布在极坐标网格上的像素. 换言之, 我们应该确定像素的位置, 使它们在极径和极角 (从子孔径的中心测量) 上以统一的增量间隔开来. 图 3 显示了与图 1 相同的数据, 但像素位于极坐标网格上. 纵轴对应极径坐标, 横轴对应极角坐标. 注意, 图 3 中距离恒定的线条是水平直线. 图 4 显示了我们的极坐标网格图像的二维 FFT. 频域数据在水平方向上只占据了一个狭窄的区域, 这意味着对图像数



据沿着横轴高度过采样. 快速后向投影利用了角度方向上粗糙的极坐标网格带宽小的优势<sup>8</sup>. 像素数量大大的减少使快速后向投影的速度比标准后向投影有了很大的提升.

我们提醒读者不要将这里所讨论的极坐标网格与著名的极坐标格式 (polar-format) SAR 成像方法相混淆. 快速后向投影中使用的极坐标网格是图像域中的极坐标网格, 而极坐标格式处理采用的是频域中的极坐标网格.

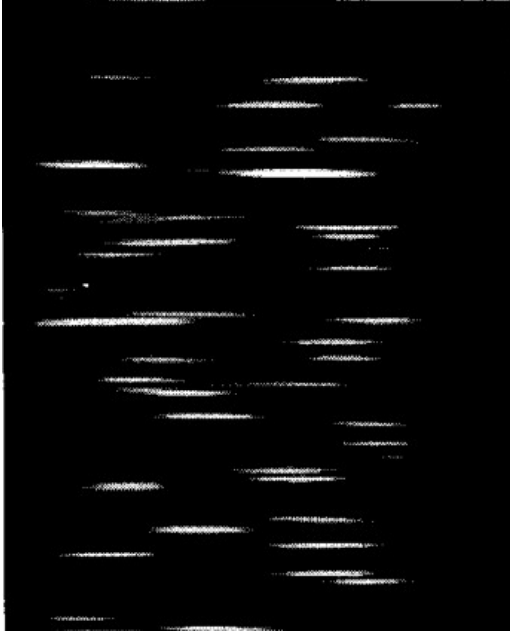


图3 位于极坐标网格上的图 1

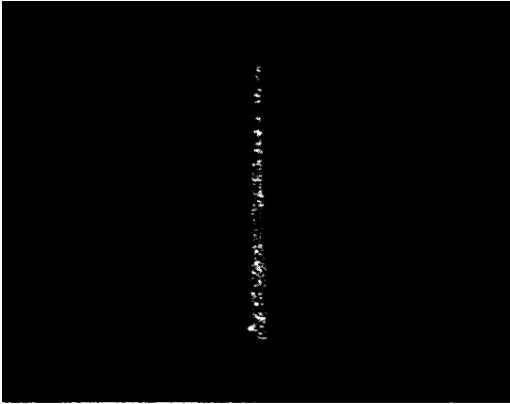


图4 图3 极坐标网格图像的二维 FFT

<sup>8</sup> 极坐标下横轴上的带宽小是指在笛卡尔坐标系中, 方位向上采样的数据带宽很大, 但变换到极坐标系下带宽却很小——相当于在极坐标系中, 在方位向上是过采样的, 这样像素数可以比原来更少, 而又达到我们想要的效果, 且计算量降低. 这也是坐标变换的精髓所在.

### C. 极坐标中的 Nyquist 采样率

在上一节中, 我们用一个简单的可视化例子来说明极坐标的带宽优势. 现在我们分析后向投影方程, 并推导出极坐标网格 (polar-grid) 图像的 Nyquist 采样率的简单表达式. 这些表达式将准确地告诉我们极坐标网格子孔径图像中的像素间距应该多大.

第  $n$  幅子孔径的成像由式 (9) 给出. 在本节中, 我们假设飞行航迹在子孔径上是一条直线; 对于足够短的子孔径来说, 这显然是一个合理的假设. 此外, 我们做这个假设只是为了计算 Nyquist 采样率; 在实际的成像过程中, 我们使用真实的飞行轨迹. 在一个子孔径里, 真正的飞行轨迹不完全是直线意味着 Nyquist 采样率可能比直线情况下的略高. 假设我们不以太接近 Nyquist 采样率的速率采样, 就不会有什么问题.

我们第一步是将成像方程重新表述为极坐标  $(r, \alpha)$  的形式. 坐标  $r$  定义为成像位置与子孔径中心的距离:

$$r = |\vec{p} - \vec{q}(s_n)|.$$

坐标  $\alpha$  定义为飞行轨迹与像素视线夹角的余弦值:

$$\alpha = \frac{1}{r} [\vec{p} - \vec{q}(s_n)] \cdot \vec{q}'(s_n).$$

对于直的飞行轨迹, 存在有用的等式

$$\begin{aligned} & [\vec{p} - \vec{q}(s_n)] \cdot [\vec{q}(s_n + \xi) - \vec{q}(s_n)] \\ &= [\vec{p} - \vec{q}(s_n)] \cdot \vec{q}(\xi) = [\vec{p} - \vec{q}(s_n)] \cdot \xi \vec{q}'(s_n) \quad (10) \\ &= \xi [\vec{p} - \vec{q}(s_n)] \cdot \vec{q}'(s_n) = r\xi\alpha. \end{aligned}$$

成像方程中的关键量是像素与孔径上的点之间的距离. 我们在极坐标中将其表示为:

$$\begin{aligned} |\vec{p} - \vec{q}(s_n + \xi)| &= |[\vec{p} - \vec{q}(s_n)] - [\vec{q}(s_n + \xi) - \vec{q}(s_n)]| \\ &= \sqrt{r^2 + \xi^2 - 2r\xi\alpha}. \end{aligned}$$

最后的等式关系来自于  $|\vec{a} - \vec{b}| = \sqrt{|\vec{a}|^2 + |\vec{b}|^2 - 2\vec{a} \cdot \vec{b}}$  和式 (10)<sup>9</sup>.

极坐标系中的子孔径成像由

$$I_n(r, \alpha) = \int_{-l/2}^{l/2} d\xi F\left(s_n + \xi, \frac{2}{c} \sqrt{r^2 + \xi^2 - 2r\xi\alpha}\right)$$

给定. 下一步是将  $F(s, t)$  展开为其傅里叶变换  $\tilde{F}(s, \nu)$  的表示:

$$I_n(r, \alpha) = \int_{-l/2}^{l/2} d\xi \int_{\nu_{\min}}^{\nu_{\max}} d\nu \tilde{F}(s_n + \xi, \nu) e^{j \frac{4\pi\nu}{c} \sqrt{r^2 + \xi^2 - 2r\xi\alpha}}.$$

<sup>9</sup> 见式 (\*).

我们最终的分析步骤是计算  $I_n(r, \alpha)$  关于  $r$  和  $\alpha$  的傅里叶变换:

$$\tilde{I}_n(k_r, k_\alpha) = \frac{1}{2\pi^2} \int_{-\infty}^{\infty} dr \int_{-\infty}^{\infty} d\alpha I_n(r, \alpha) e^{-j2\pi(k_r r + k_\alpha \alpha)}.$$

用驻定相位法 (stationary phase method) [17, 33 页] 可以相当准确地计算出变换结果. 驻定相位条件为

$$\frac{\partial}{\partial \alpha} \left( \frac{4\pi\nu}{c} \sqrt{r^2 + \xi^2} - 2r\xi\alpha - 2\pi k_\alpha \alpha \right) = 0$$

和

$$\frac{\partial}{\partial r} \left( \frac{4\pi\nu}{c} \sqrt{r^2 + \xi^2} - 2r\xi\alpha - 2\pi k_r r \right) = 0.$$

求解上述方程得, 当满足

$$-\frac{\nu_{\max} l}{c} \leq k_\alpha \leq \frac{\nu_{\max} l}{c}$$

和

$$\frac{2\nu_{\min}}{c} \leq k_r \leq \frac{2\nu_{\max}}{c}$$

的时候,  $\tilde{I}_n(k_r, k_\alpha)$  是非零的. 在写上面两个方程的时候, 我们做了一个额外的近似, 即子孔径长度远小于距离 ( $l \ll r$ ). 这个近似值不需要用到, 但它几乎适用于所有感兴趣的情况, 它也使结果特别地清楚.

对  $k_r$  和  $k_\alpha$  的约束可以直接转化为对  $r$  和  $\alpha$  的 Nyquist 采样要求:

$$\Delta\alpha \leq \frac{c}{2\nu_{\max} l}, \quad (11)$$

$$\Delta r \leq \frac{c}{2(\nu_{\max} - \nu_{\min})}. \quad (12)$$

$\alpha$  所需的采样数显然与子孔径长度  $l$  成正比. 另一方面,  $r$  的采样率只取决于系统带宽, 与子孔径尺寸无关.

## D. 实现

到目前为止, 我们主要关注的快速后向投影的推导和数学问题. 现在我们将描述该算法的合理实现. 与标准后向投影一样, 我们将逐脉冲地处理数据: 读取一个脉冲, 计算它对图像的贡献, 然后转到下一个脉冲. 式 (2) 中的脉冲滤波可以用与标准后向投影相同的方式进行: 用时域 FIR 滤波器或者简单地使用 FFT.

第一步是为所需的输出图像选择一个成像网格. 这意味着我们必须知道图像中每个像素的三维坐标, 而且我们必须分配足够的存储空间 (内存或磁盘空间) 来存

放每个像素的复数值. 如果像素被安排在一个规则的笛卡尔 (Cartesian) 网格中, 只需要指定角点像素的位置以及距离向和横向的步进位移向量即可. 如果像素位于不规则的位置 (例如, 在一个真实地形的网格上), 那么我们要存储每个像素的三维位置.

接下来, 我们必须选择子孔径尺寸. 正如下一节所示, 对于  $N \times N$  像素的图像来说, 最佳的子孔径尺寸大约包含  $\sqrt{N}$  个脉冲. 在实践中, 通常可以通过少量的试错 (trial-and-error) 测试来选择一个合适的孔径尺寸.

处理过程自然地以逐子孔径 (subaperture-by-subaperture) 为基础. 对于每个子孔径, 我们计算以当前子孔径中心为原点的极坐标下的像素位置. 极径和极角方向上的像素间距选择略高于式 (11) 和 (12) 的 Nyquist 采样率<sup>10</sup>. 正如前面所说, 像素间距在角度方向上是非常稀疏的.

子孔径的脉冲处理与标准后向投影完全一样, 只是像素位于特殊的极坐标网格上. 特别地是, 我们使用子孔径上真实的运动历程 (motion history), 而不是直线近似值.

一旦我们完成了对当前子孔径脉冲的处理, 我们必须将我们的结果从极坐标网格转化回最终输出的成像网格, 这需要几个步骤. 首先, 我们从极坐标图像数据中去除载波相位. 接下来, 我们在两个方向上对极坐标图像数据上采样<sup>11</sup>. 然后, 我们通过在上采样的极坐标图像数据中找到相应的点, 计算出对输出图像中每个像素的贡献. 二维线性插值被用来近似上采样极坐标网格上像素之间的极坐标图像数据. 在线性插值后, 恢复载波相位. 从极坐标网格转化到输出网格的准确度完全取决于上采样系数. 上采样越多, 结果越准确. 在实践中, 2 到 8 的上采样系数比较合理地兼顾了速度和精度<sup>12</sup>.

在极坐标网格图像转换到输出图像网格后, 我们转到下一个子孔径, 并计算一个新的极坐标网格. 我们继续处理子孔径, 直到所有的脉冲都被处理完. 由于输出图像数据每一个子孔径更新一次, 因此将输出图像存储到磁盘而非内存中不是没有道理的. 这个程序允许用相对较少的内存来创建非常大的图像, 而在速度上仅有轻微的损失. 极坐标网格图像必须存储在内存中, 但它们所需的

<sup>10</sup> 一方面是保证满足采样要求; 另一方面是避免成像点脱离网格 (off grid).

<sup>11</sup> 见脚注 6.

<sup>12</sup> 应该记住, 我们是对稀疏的极坐标网格图像做上采样, 它所包含的像素远远少于最终的图像. 稀疏性可以防止上采样使算法的速度太慢.

$$\begin{aligned} |\bar{p} - \bar{q}(s_n + \xi)| &= |[\bar{p} - \bar{q}(s_n)] - [\bar{q}(s_n + \xi) - \bar{q}(s_n)]| \\ &= \sqrt{|\bar{p} - \bar{q}(s_n)|^2 - |[\bar{q}(s_n + \xi) - \bar{q}(s_n)]|^2 - 2[\bar{p} - \bar{q}(s_n)] \cdot [\bar{q}(s_n + \xi) - \bar{q}(s_n)]} \\ &= \sqrt{r^2 + \xi^2 - 2r\xi\alpha} \end{aligned} \quad (*)$$

空间要小得多 (即使是在上采样之后), 因为它们在角度方向上是稀疏的.

#### E. 计算负荷和最佳子孔径尺寸

假定我们想用  $N_{pulse}$  个脉冲的距离压缩数据在聚束模式下生成一个  $N \times N$  像素的图像. 假设每个子孔径包含  $m$  个脉冲, 那么有  $N_{pulse}/m$  个子孔径需要处理. 每个子孔径的极坐标网格包含  $Nm$  个像素. 对于每个子孔径, 我们必须执行下面的任务, 括号里是相应的计算量:

1. 计算极坐标网格坐标 ( $Nm$ ).
2. 在极坐标网格上做后向投影 ( $Nm^2$ ).
3. 对极坐标网格上采样 ( $Nm$ ).
4. 将极坐标网格插值为笛卡尔输出网格 ( $N^2$ ).

整幅图像总的计算量为

$$\begin{aligned}
 N_{op} &\approx N_{pulse} N \left( 1 + m + \frac{N}{m} \right) \\
 &\geq N_{pulse} N \left( 1 + 2\sqrt{m \cdot \frac{N}{m}} \right) \\
 &= N_{pulse} N \left( 1 + 2\sqrt{N} \right) \\
 &\approx N_{pulse} N \cdot \sqrt{N} = N^{3/2} N_{pulse}.
 \end{aligned} \tag{13}$$

通过选择  $m = \sqrt{N}$  最小化计算量. 如果我们使子孔径变小, 我们就不得不非常频繁地更新完整的高分辨率图像. 如果我们把子孔径变大, 子孔径图像的横向分辨率就会过高. 最佳的子孔径大小代表了这两种影响之间的折中. 只保留主要项, 我们看到快速后向投影的计算量规模为

$$N_{op} \approx N^{3/2} N_{pulse}.$$

与式 (6) 相比表明, 快速后向投影比标准后向投影要快  $\sqrt{N}$  倍.

#### IV. 比较快速后向投影与标准后向投影

为了补充上一节的理论分析, 我们现在展示标准后向投影与快速后向投影数值上的比较. 为了演示脉冲响应 (impulse response), 我们使用一个理想点散射体 (即在频率和孔径上的响应都是均匀的) 的仿真合成数据. 数据在频率和孔径上都加了一个 Kaiser 窗. 不加窗的话, 标准和快速头像投影之间的差异将隐藏在脉冲响应的旁瓣中.

基本仿真参数如下:

- 带宽: 20 MHz 到 90 MHz
- 相干积累角:  $-45^\circ$  到  $+45^\circ$

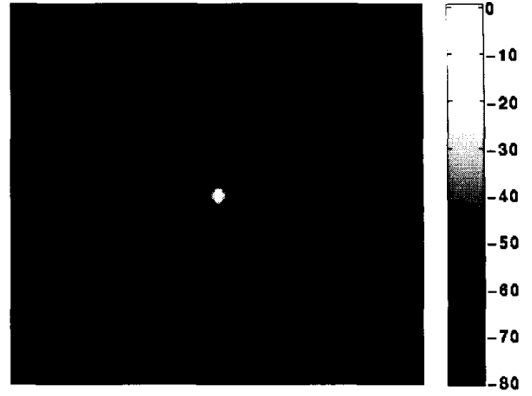


图5 标准后向投影, 上采样系数 = 12

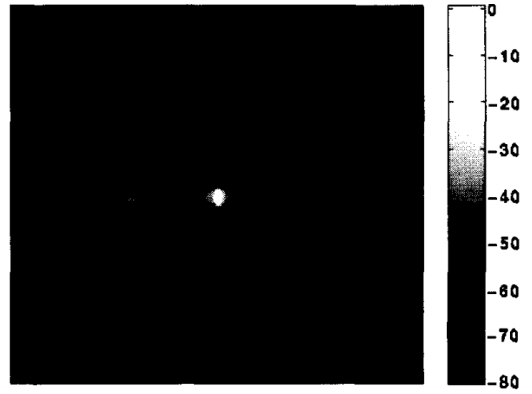


图6 快速后向投影, 上采样系数 = 1

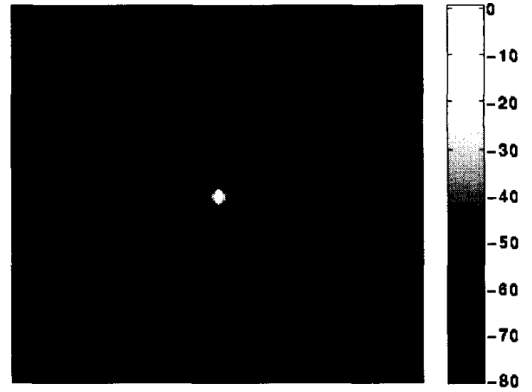


图7 快速后向投影, 上采样系数 = 8

- 像素尺寸: 距离向 1.5 m  $\times$  横向 1.0 m
- 图像尺寸: 2000  $\times$  2000 像素
- 图像中心: 距离飞行轨迹 4000 m

标准后向投影程序在这个数据上运行了两次. 在第一次运行中, 我们对距离脉冲进行 6 倍的上采样; 这次运行是用来确定标准后向投影的运算速度<sup>13</sup>. 在第二次运行中, 我们对距离脉冲进行 12 倍的上采样; 这次运行的图像被用作检测快速后向投影成像质量的参考图像. 快

<sup>13</sup> 标准后向投影的速度对于高达 6 的上采样系数来说基本是恒定的.

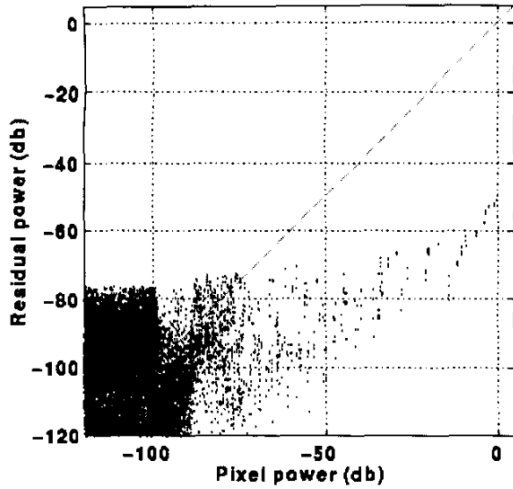


图8 快速后向投影残差, 上采样系数 = 8

速后向投影也运行了若干次, 使用的是 1 到 8 的上采样系数. 每种情况下都通过选择合适的子孔径尺寸优化处理速度.

#### A. 成像质量

图 5 显示了使用标准后向投影所成的  $200 \times 200$  像素的部分图像. 该图像经过归一化处理, 使峰值像素处于 0 dB. 图中显示了一个宽的动态范围 (80 dB), 以强调在主瓣 (main peak) 附近没有信号处理导致的伪影 (artifact). 旁瓣已经被频率和孔径上的 Kaiser 窗压制.

图 6 和图 7 显示了使用快速后向投影所成的图像, 上采样系数为 1 和 8. 同样, 我们在图像中使用宽广的动态范围以使信号处理的伪影明显. 请注意, 随着上采样系数从 1 到 8 的增加, 成像质量有所改善.

为了从数值上评估快速后向投影的质量, 我们将每个图像与图 5 中的标准后向投影进行比较. 这种比较是从标准后向投影成像相干减去每个快速后向投影成像来实现的. 图 8 显示了标准后向投影成像在减去 8 倍上采样的快速后向投影成像后的残差像素功率散点图. 横轴表示标准后向投影成像中的像素功率, 而纵轴表示相干相减后的对应像素的功率. 对角线表示误差变得和正确像素值一样大的边界.

#### B. 处理速度

衡量一个给定后向投影算法速度的一个合理方式衡量每秒的脉冲像素 (pulse-pixel) 数. 换句话说, 将  $N$  个脉冲处理成一幅  $M$  个像素的图像需要多长的时间. 对于标准后向投影, 答案只取决于  $NM$ . 因此, 我们以每秒百万个脉冲像素 (Mpp/s) 来表示算法的速度.

标准后向投影的运行速度为 0.71 Mpp/s. 对于快速后向投影, 处理速度取决于上采样系数. 表 1 显示了处理速度 (单位: Mpp/s)、相对于标准后向投影的速度以及最大相干残差功率 (单位: dB) 的关系:

表 1 快速后向投影与标准后向投影的数值比较

上采样系数	速度	相对速度	最大残差
1	20.2	28.5	-13.7
2	18.2	25.6	-25.5
3	15.3	21.5	-32.0
4	11.7	16.5	-37.5
6	7.2	10.1	-45.2
8	4.8	6.8	-50.1

最大的残差总是出现在主瓣峰值上. 例如, 在上采样系数为 4 的情况下, 最大的残差功率为 -37.5 dB, 这意味着主瓣峰值处的像素复数值有 1.3% 的误差. 远离主瓣的残差总是要小得多.

## V. 结论

我们提出了一种快速后向投影算法, 该算法与标准后向投影的结果相同, 但在现实情况下运行速度要快 10 到 30 倍. 通过改变升采样系数, 用户可以根据特定应用的需要, 以速度换取精度. 该算法可以应用于任何类型的合成孔径雷达数据, 并取得良好的效果, 特别是对存在大量偏轨运动的宽频、广角和/或具有长相干孔径的传感器处理方式. 它也将是一种处理在非线性轨迹上采集的数据的方法.

## 致谢

作者要感谢 Serpil Ayasli 和 Irvin Stiglitz 对本文研究的支持和鼓励. 还要感谢 Keith Forsythe、Michael Touns 和 Gerald Benitz 就技术问题进行的有益讨论.

这项工作是由国防高级研究计划局 (Defense Advanced Research Projects Agency) 根据空军合同 F19628-95-C-0002 赞助的. 意见、见解、结论和建议代表作者个人观点, 不一定与美国政府一致.

## 参考文献

### 原文参考文献

- [1] W. M. Brown and R. J. Fredericks, "Range-Doppler Imaging with Motion through Resolution Cells", *IEEE*

- Transactions on Aerospace and Electronics Systems*, vol. AES-5, no. 1, pp. 98–102, January 1969.
- [2] D. C. Munson and R. L. Visentin, “A Signal Processing View of Strip-Mapping Synthetic Aperture Radar”, *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. 37, no. 12, pp. 2131–2147, December 1989.
- [3] G. Carrara, R. S. Goodman and R. M. Majewski, *Spotlight Synthetic Aperture Radar: Signal Processing Algorithms*, Artech House, 1995.
- [4] R. H. Stolt, “Migration by Fourier Transform”, *Geophysics*, vol. 43, no. 1, pp. 23–48, February 1978.
- [5] F. Rocca, C. Cafforio and C. Prati, “Synthetic Aperture Radar: A New Application for Wave Equation Techniques”, *Geophysical Prospecting*, vol. 37, pp. 809–830, 1989.
- [6] C. Cafforio, C. Prati and F. Rocca, “SAR Data Focusing Using Seismic Migration Techniques”, *IEEE Transactions on Aerospace and Electronic Systems*, vol. 27, no. 2, pp. 194–206, March 1991.
- [7] M. Soumekh, “A System Model and Inversion for Synthetic Aperture Radar”, *IEEE Transactions on Image Processing*, vol. 1, no. 1, pp. 64–76, January 1992.
- [8] A. S. Milman, “SAR Imaging by  $\omega$ - $k$  Migration”, *International Journal of Remote Sensing*, vol. 14, no. 10, pp. 1965–1979, 1993.
- [9] R. K. Raney and P. W. Vachon, “A Phase Preserving SAR Processor”, in *1989 IEEE Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 2588–2591, July 1989.
- [10] H. Runge and R. Bamler, “A Novel High Precision SAR Focussing Algorithm Based on Chirp Scaling”, *Proceedings of the 1992 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 372–375, May 1992.
- [11] R. K. Raney, H. Runge, R. Bamler, I. G. Cumming and F. H. Wong, “Precision SAR Processing Using Chirp Scaling”, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 32, no. 4, pp. 786–799, July 1994.
- [12] H. Hellsten, L. M. H. Ulander, A. Gustavsson and B. Larsson, “Development of VHF CARABAS II SAR”, in *Proc. SPIE - Int. Soc. Opt. Eng. (USA)*, vol. 2747, pp. 48–60, 1996.
- 附加参考文献
- [13] Object Management Group, Lincoln Laboratory, Massachusetts Institute of Technology. *Corner Turn Benchmark*.
- [14] The Five Fearless Thinkers, *Image Processing Using SPECT Analysis, Backprojection Filters*, 1986.
- [15] Alan V. Oppenheim, Alan S. Willsky, S. Hamid Nawab, *Signals and Systems, Second Edition*, PRENTICE HALL, 1983.
- [16] Mark A. Richards, *Fundamentals of Radar Signal Processing, Second Edition*, McGraw-Hill, 2014.
- [17] 张晓玲, 师君, 韦顺君. 三维合成孔径雷达 [M]. 北京: 国防工业出版社, 2017.