文章编号:1000-582X(2012)09-099-07

数字多波束逐点聚焦延时参数的压缩算法

王 平,范文政,高 阳,许 琴,何 为
(重庆大学 输配电装备及系统安全与新技术国家重点实验室,重庆 400044)

摘 要:针对数字多波束合成技术在超声成像领域应用中存在逐点聚焦延时参数的存储容量过大的问题,提出了一种四波束逐点聚焦延时参数的压缩存储与实时生成算法。首先将四波束的聚焦延时问题转换成单波束的聚焦延时问题,然后对转换后的单波束聚焦延时数据进行量化、压缩和存储。在聚焦时,将压缩存储的延时参数进行解压,实时生成四波束所需的延时参数。以 8 通道 128 阵元的平阵探头为例,对该算法进行了相关的数学推导和证明,并对其性能进行了分析讨论,验证了该算法的优越性。

关键词:波束合成;聚焦延时参数;单波束合成;压缩存储;实时生成
中图分类号:TB553 **文献标志码**:A

The compression algorithm for focusing delay data in digital multi-beam forming

WANG Ping, FAN Wenzheng, GAO Yang, XU Qin, HE Wei
(State Key Laboratory of Power Transmission Equipment & System Security and New Technology, Chongqing University, Chongqing 400044, China)

Abstract: Digital multi-beam forming has been widely used in ultrasound imaging system. It can effectively improve frame rate and image resolution. However, the mass of focusing delay data makes it impractical to directly implement dynamic focusing. A compression algorithm for four-beam focusing delay data is proposed to effectively reduce the storage of focusing delay data. Firstly, the calculation of four-beam focusing delay data is simplified to that of single-beam focusing delay data, which are then quantified, compressed and stored. When in dynamic receive focusing process, focusing delay data can be generated by decompressing the stored data for each channel. Take 8-channel 128-element linear array of transducers as an example, the design procedures and related mathematical derivation are described in detail and performance of the algorithm is discussed to verify the algorithm's superiority.

Key words: beam forming; focusing delay data; single-beam forming; compression; real-time generating

临床医学设备大多要求数字超声影像系统能够具有高帧频的实时成像性能,但是在传统的单波束合成技术中,图像的帧频受到超声传播速度和探测深度的限制,往往无法满足许多疾病诊断的要求。

在超声成像系统领域,波束合成技术长期以来一直都是研究的热点[1]。在保证图像质量的前提下,提高图像帧频的方法往往是采用数字多波束合成技术[2-3]。该技术主要是通过特殊设计的发射–接

收稿日期:2012-04-25
基金项目:广东省教育部产学研结合项目(2008B090500272);第四届国家大学生创新性实验项目(101061117)
作者简介:王平(1976-),男,重庆大学副教授,博士,主要从事数字超声影像系统的研究,(E-mail)cqu_dq@163.com。

收方式，利用一次发射合成 N 根波束[4]，理论上便可将帧频提高 N 倍。然而在这个过程中，聚焦延时参数的存储量将显著扩大。

为了实现高精度的逐点聚焦，必须解决聚焦延时参数实时生成的问题[5-7]。目前主要的解决方案有2种：一种是实时计算，另一种是对聚焦延时参数进行存储。如果各个阵元的延时参数均采用实时计算的方法，那么该方法虽然减小了数据的存储量，但计算过程中涉及复杂的乘法运算等；如果直接对聚焦延时参数进行存储，那么将会导致存储量非常庞大。哈尔滨工业大学提出了一种医学超声成像系统中的数字波束形成的聚焦参数压缩存储方法，该方法根据波束形成聚焦参数的变化趋势，在波束形成聚焦延时参数的输入存储中增加了游程编码技术和预测编码技术[8]，但是该方法对存储量的需求仍然较大。在一些国外的参考文献中，也有采用中点算法[9]，通过实时计算，生成聚焦延时参数，但是对硬件资源的消耗非常大。

通过对超声多波束回波信号聚焦延时路径的特点和变化规律进行深入分析，笔者提出了一种四波束逐点聚焦参数的压缩存储算法。该算法利用超声回波路径平行移动的几何方法，将四波束聚焦延时参数的计算转换成单波束聚焦延时参数的计算。根据每个通道逐点聚焦延迟时间函数 $\tau_i(F,\beta_i)$ 的特点，将各个通道的聚焦延时时间转换成相邻通道聚焦延时时间依次累加的关系，量化后进行压缩存储。在逐点聚焦的过程中，实时解压并生成4条波束聚焦线所需的聚焦延时参数，从而实现了高精度的四波束逐点聚焦。该逐点聚焦方法在大幅度降低各个通道延时数据存储量的同时，也避免了复杂的乘法运算。

1 数字四波束逐点聚焦延时数据表压缩存储算法

1.1 四波束聚焦延时参数的计算

超声接收聚焦的本质就是对不同通道接收到的超声回波信号按其路径施加特定的延时，然后再相加求和得到目标点的聚焦信号[10-12]。因此，准确计算出聚焦延迟时间是逐点聚焦的关键[13-14]。笔者以8通道128阵元平阵探头为例来说明四波束逐点聚焦延时参数的计算，如图1所示。

在图1中，4条实线1、2、3、4为4条波束扫描线，虚线0为中心对称线。添加虚拟点将每个阵元平分为4等份，另在8通道阵元的两侧等距离(即 d_0)处各添加一个虚拟点，总共有35个虚拟点。超声回波信号接收点为每个阵元的中心点。

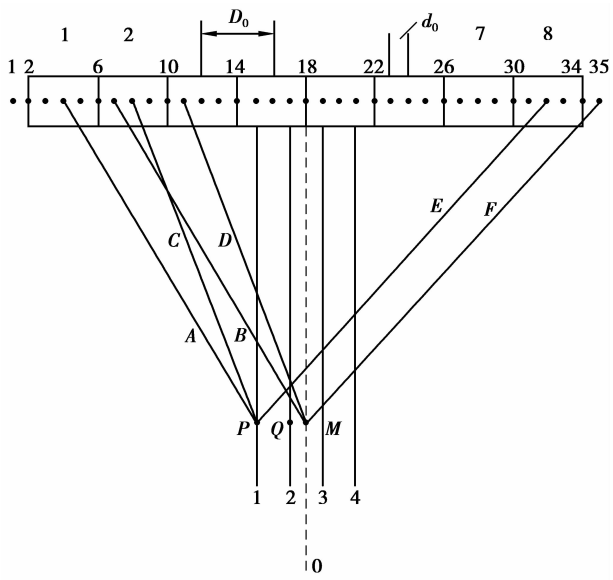

图1 8通道平阵探头聚焦示意图

假设阵元间距 D_0 为0.48 mm，则虚拟点间距 $d_0=D_0/4=0.12$ mm，子阵阵元数 k 为8。在图1中，M 点为0号线上任意一点，P 点为1号线上一点，Q 点为2号线上一点，M 点、P 点和 Q 点到平阵探头的垂直距离相等。

不难看出，P、Q 点到1～8号阵元的聚焦延时时间为

$$\begin{cases}\tau_P=\{\tau_{P,4},\tau_{P,8},\cdots,\tau_{P,32}\},\\ \tau_Q=\{\tau_{Q,4},\tau_{Q,8},\cdots,\tau_{Q,32}\}。\end{cases}\tag{1}$$

P 点到1阵元的距离 A(即 P 点到4号虚拟点的距离)等于 M 点到7号虚拟点的距离 B；P 点到2阵元的距离 C(即 P 点到8号虚拟点的距离)等于 M 点到11号虚拟点的距离 D；P 点到8阵元的距离 E(即 P 点到32号虚拟点的距离)等于 M 点到35号虚拟点的距离 F。因此，P 点到8个阵元的距离分别等于 M 点到7、11、15、19、23、27、31和35号虚拟点的距离，即

$$\tau_P=\tau_M=\{\tau_{M,7},\tau_{M,11},\cdots,\tau_{M,35}\}。\tag{2}$$

同理，2号扫描线上 Q 点到8个阵元的距离分别等于 M 点到5、9、13、17、21、25、29和33号虚拟点的距离，即

$$\tau_Q=\tau_M=\{\tau_{M,5},\tau_{M,9},\cdots,\tau_{M,33}\}。\tag{3}$$

根据以上分析可以得出：1、2号扫描线聚焦延时参数的计算可以转化为0号线对特定虚拟点的聚焦延时参数的计算。3、4号扫描线与2、1号扫描线关于中线(即0号线)对称，因此只需要考虑1、2号

扫描线的聚焦延时参数。

利用上述的平移方法，只需计算 M 点到 5、7、9、11、13、15、17、19、23、25、27、29、31、33 和 35 号虚拟点的延时参数，即可以得到 4 条扫描线所需的聚焦延时参数。由于所有虚拟点关于 18 号点对称，因此 4 条扫描线聚焦延时参数的计算就简化为计算 M 点到 1、3、5、7、9、11、13、15 和 17 号点的延时参数，即

$$\tau = \{\tau_{M,1}, \tau_{M,3}, \cdots, \tau_{M,17}\}。\tag{3}$$

令图 2 中 $d = D_0/2 = 0.24$ mm，并且图 2 中 N 点到平阵中心的距离与图 1 中 M 点到平阵中心的距离相等，因此图 1 中 1、3、5、7、9、11、13、15 和 17 点到 M 的距离与图 2 中 1～9 阵元到 N 点的距离分别相等，即有

$$\tau_{M,2i-1} = \tau_{N,i}, i = 1,2,\cdots,9。$$

因此，对图 1 中四波束聚焦延时参数的计算可以转换为对图 2 中单波束聚焦延时参数的计算。

在图 2 中，设 F 为焦距（从子阵中心到焦点 N 的距离），探测深度 L 为 240 mm，超声波在人体软组织中的平均速度 c 为 1 540 m/s，相邻阵元之间的距离为原阵元间距的 1/2，$d = 0.24$ mm，可以得出第 i 阵元与聚焦中心线之间的距离 a_i 为

$$a_i = \left|\frac{18+1}{2} - i\right| \times d, (i = 1,2,\cdots,18)。\tag{4}$$

由式(4)可得

$$0.12\ \text{mm} \leqslant a_i \leqslant 2.04\ \text{mm}。$$

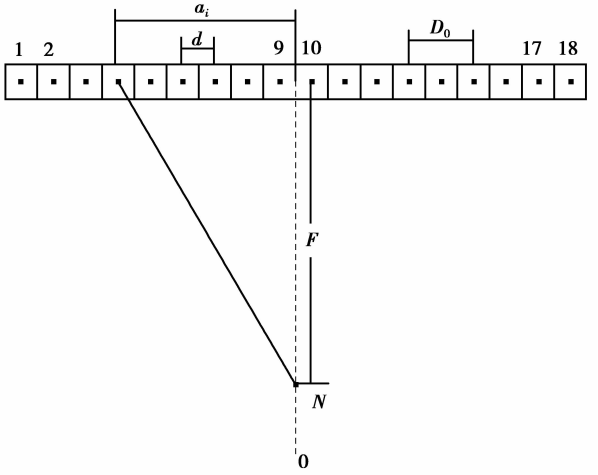

图 2　8 通道平阵探头聚焦等效转换示意图

根据勾股定理，可以计算出第 i 阵元的延时时间为

$$\tau_i(F,a_i) = (\sqrt{a_i^2 + F^2} - F)/c。\tag{5}$$

设 AD 转换器速度 f_s 为 50 MHz，实现逐点聚焦要求对声束轴线上的每一个回波点都进行聚焦，因此聚焦线上聚焦点的间隔 ΔF 为[15]

$$\Delta F = c/2f_s = 0.015\ 4\ \text{mm}。\tag{6}$$

采用 18 阵元的中心对称接收方式，根据式(4)～(6)可以计算出 18 通道的逐点聚焦延迟时间，考虑到接收通道左右中心对称，仅列出聚焦深度从 2～240 mm之间 1～9 通道逐点聚焦延时参数，如表 1 所示。

表 1　1～9 通道的逐点聚焦延时参数　　ns

序号	深度/mm	聚焦通道								
		1	2	3	4	5	6	7	8	9
1	2.000 0	556.40	448.52	348.35	257.36	177.25	109.90	57.18	20.87	2.34
2	2.015 4	553.41	445.97	346.24	255.71	176.06	109.12	56.76	20.71	2.32
3	2.030 8	550.45	443.44	344.16	254.09	174.88	108.36	56.35	20.56	2.30
					……					
15 453	239.960 8	5.63	4.38	3.29	2.36	1.58	0.95	0.49	0.18	0.02
15 454	239.976 2	5.63	4.38	3.29	2.36	1.58	0.95	0.49	0.18	0.02
15 455	239.991 6	5.63	4.38	3.29	2.36	1.58	0.95	0.49	0.18	0.02

从表 1 可以看出：如果对表中数据取整后直接存储，那么存储容量将会非常庞大。

1.2　聚焦延时参数的压缩存储算法

考察逐点聚焦延时数据的变化特点，对 $\tau_i(F, a_i)$ 的表达式求导，可得

$$\begin{cases} \dfrac{\mathrm{d}\tau_i(F,a_i)}{\mathrm{d}F} = \left[\dfrac{F}{\sqrt{a_i^2+F^2}} - 1\right]/c \leqslant 0, \\ \dfrac{\mathrm{d}^2\tau_i(F,a_i)}{\mathrm{d}F_i^2} = \left[\dfrac{\sqrt{a_i^2+F^2} - \dfrac{F^2}{\sqrt{a_i^2+F^2}}}{a_i^2+F^2}\right]/c \geqslant 0, \\ \dfrac{\mathrm{d}\tau_i(F,a_i)}{\mathrm{d}a_i} = \left[\dfrac{a_i}{\sqrt{a_i^2+F^2}}\right]/c \geqslant 0。 \end{cases}\tag{7}$$

根据式(7)可知，$\tau_i(F,a_i)$是 F 的单调递减凹函数，其递减速度逐渐变慢。$\tau_i(F,a_i)$是 a_i 的单调递增函数，因此是 i 的单调递减函数，$\tau_i(F,a_i)$函数如图 3 所示。

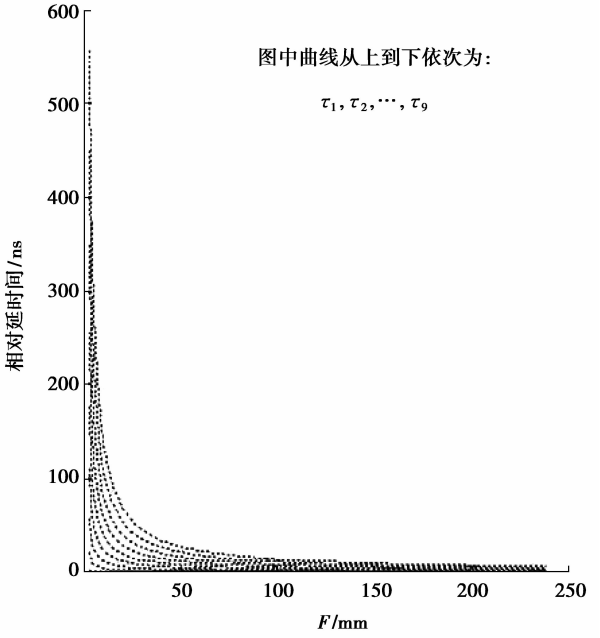

图 3 聚焦线的焦点到各个阵元延时 $\tau_i(F,a_i)$

结合图 3 和表 1 可以发现：每个通道逐点聚焦的延时时间 $\tau_i(F,a_i)$随着聚焦深度 F 的增加逐渐减小，$\tau_i(F,a_i)$函数的梯度变化也相应减小，延时时间变化速度也逐渐减慢。靠近聚焦线的第 9 阵元，聚焦延时时间 $\tau_9(F,a_9)$变化小、变化慢，对存储容量需求较小；而远离聚焦线的第 1 阵元，聚焦延时时间 $\tau_1(F,a_1)$的数值变化范围大、变化速度快，这是造成对存储容量需求大的根本原因。

因此为了实现聚焦延时参数的压缩存储，必须缩小聚焦延时时间的数据变化范围，减小聚焦延时参数变化的速率。根据式(5)，可以计算得到相邻阵元之间的延时时间 $\Delta\tau_i(F,a_i)$。

$$\begin{cases}\Delta\tau_i(F,a_i)=\tau_i(F,a_i)-\tau_{i+1}(F,a_i),\\(i=1,2,\cdots,8);\\\Delta\tau_9(F,a_9)=\tau_9(F,a_9)。\end{cases}\tag{8}$$

$\Delta\tau_i(F,a_i)$仍然是 F 与 i 的单调减函数，随着 F 与 i 的增加，$\Delta\tau_i(F,a_i)$减小，1～9 阵元的相对延时时间 $\Delta\tau_i(F,a_i)$函数如图 4 所示。

根据三角形任意两边之差小于第三边的原则，式(8)中 $\Delta\tau_i(F,a_i)$的取值范围满足以下约束条件：

$$0<\Delta\tau_i(F,a_i)<\frac{d}{c}。\tag{9}$$

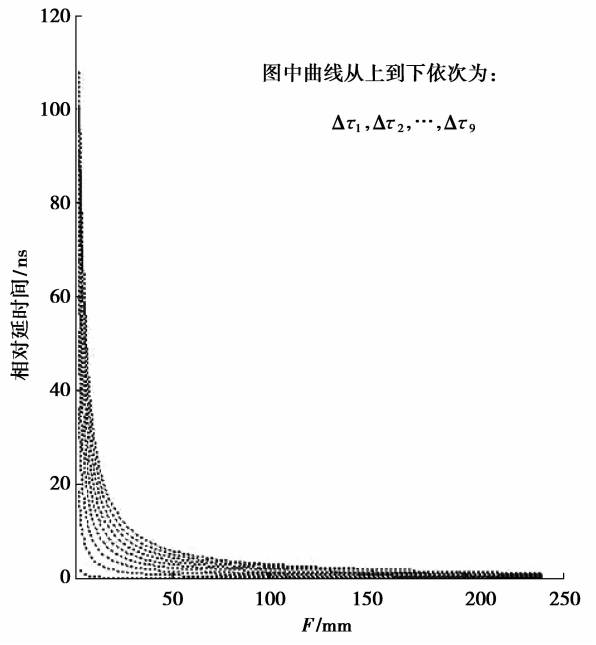

图 4 聚焦线上的焦点到各个阵元的相对延时时间 $\Delta\tau_i(F,a_i)$

因此只要确定了第 i 阵元的聚焦延时时间 $\tau_i(F,a_i)$，相邻的第 $i-1$ 阵元的聚焦延时时间可以根据式(8)递推求出。由于 $\Delta\tau_i(F,a_i)$满足约束条件式(9)，在同等精度的量化条件下，对 $\Delta\tau_i(F,a_i)$的存储比 $\tau_i(F,a_i)$相对容易，存储量也较小。

根据以上分析，τ_i 为第 i 阵元的聚焦延时时间，$\Delta\tau_i$ 为第 i 阵元相对于第 $i+1$ 阵元的聚焦相对延时时间，则 $\Delta\tau_i$ 与 τ_i 存在如下关系：

$$\begin{aligned}&\tau_9=\Delta\tau_9,\\&\tau_8=\Delta\tau_8+\tau_9,\\&\cdots\cdots\\&\tau_2=\Delta\tau_2+\tau_3,\\&\tau_1=\Delta\tau_1+\tau_2。\end{aligned}\tag{10}$$

根据以上分析，对表 1 的数据采用式(8)进行分解，可计算出 $\Delta\tau_i$，从而得到 1～9 通道的逐点聚焦相对延时参数，如表 2 所示。

将表 2 与表 1 的数据进行比较，可见 2～240 mm之间，1～9 通道阵元的逐点聚焦相对延时参数具有以下特征：

1)表 2 中各通道延时数据的初始值远小于表 1 中数据的初始值，表 2 中的数值变化范围明显减小，并且递减速度明显减慢。

2)根据表 2 中的数据，应用式(10)可以实时计算生成表 1 数据。

因此，根据表 2 数据的特征(1)可以实现聚焦延时参数的压缩存储，根据表 2 的特征(2)能够完成压缩参数的实时计算。

表 2　1～9 通道的逐点聚焦相对延时参数　　ns

深度/mm	通道								
	1	2	3	4	5	6	7	8	9
2.000 0	107.87	100.18	90.99	80.10	67.36	52.71	36.31	18.54	2.34
2.015 4	107.44	99.72	90.53	79.65	66.94	52.36	36.05	18.40	2.32
2.030 8	107.01	99.28	90.07	79.21	66.53	52.01	35.79	18.26	2.30
……									
239.960 8	1.25	1.09	0.94	0.78	0.62	0.47	0.31	0.16	0.02
239.976 2	1.25	1.09	0.94	0.78	0.62	0.47	0.31	0.16	0.02
239.991 6	1.25	1.09	0.94	0.78	0.62	0.47	0.31	0.16	0.02

1.3　聚焦延时参数的压缩存储

从图 4 可知,当 F 从 2 mm 到 240 mm 变化过程中,$\Delta\tau_i(F,a_i)$ 逐步减小,当聚焦点从 F 变化到 $F+\Delta F$ 时,$\Delta\tau_i(F,a_i)$ 的变化量记为 $\Delta\gamma$。

$$\Delta\gamma = \Delta\tau_i(F,a_i) - \Delta\tau_i((F+\Delta F),a_i)。\quad (11)$$

当 F 从 2 mm 到 240 mm 变化过程中,令 $\Delta\tau_i(F,a_i)$ 的最大变化量记为 $\Delta\gamma_{max}$。

$$\Delta\gamma_{max} = \max[\Delta\tau_i(F,a_i) - \Delta\tau_i((F+\Delta F),a_i)],\quad F\in[2,240],1\leqslant i\leqslant 8。\quad (12)$$

应用 Matlab 进行数值计算方法求解,可以求出 $\Delta\gamma_{max}=0.459\ 3$ ns,结合工程实际的需要,为了实现高精度的逐点聚焦,设各个阵元的聚焦延时量化因子为 γ,对 $\Delta\tau_i(F,a_i)$ 量化后取整的值记为 $\Delta T_i(F,a_i)$。

$$\Delta T_i(F,a_i) = \left[\frac{\Delta\tau_i(F,a_i)+\gamma/2}{\gamma}\right],\quad (i=1,2,\cdots,9)。\quad (13)$$

根据式(13)对表 2 数据进行量化取整,将各个通道的聚焦相对延时数据全部量化为整数。当满足 $\Delta\gamma_{max}<\gamma$ 约束条件时,可以得出关系式(14)。

$$\frac{\Delta\gamma}{\gamma} = \frac{\Delta\tau_i(F,a_i) - \Delta\tau_i((F+\Delta F),a_i)}{\gamma} < 1。\quad (14)$$

根据式(13)、式(14)可以推导出式(15)。

$$\begin{cases} |\Delta T_i(F,a_i) - \Delta T_i(F+\Delta F,a_i)| = 1, \\ 或 \\ |\Delta T_i(F,a_i) - \Delta T_i(F+\Delta F,a_i)| = 0。 \end{cases}\quad (15)$$

因此只要用大于 $\Delta\gamma_{max}$ 的聚焦延时量化因子 γ 对表 2 中数据 $\Delta\tau_i(F,a_i)$ 进行量化取整,可得到 $\Delta T_i(F,a_i)$,即可保证当聚焦点 F 从 2 mm 到 240 mm变化过程中,$\Delta T_i(F,a_i)$ 的变化小等于 1。

令 γ=2.5ns>$\Delta\gamma_{max}$,根据式(13),对表 2 中的延时参数进行量化取整处理后,可以得到表 3。

表 3　量化取整后的 1～9 通道的逐点聚焦相对延时参数　　2.5 ns

深度/mm	通道								
	1	2	3	4	5	6	7	8	9
2.000 0	43	40	36	32	27	21	15	7	1
2.015 4	43	40	36	32	27	21	14	7	1
2.030 8	43	40	36	32	27	21	14	7	1
……									
239.960 8	0	0	0	0	0	0	0	0	0
239.976 2	0	0	0	0	0	0	0	0	0
239.991 6	0	0	0	0	0	0	0	0	0

从表 3 中可以看出,每个通道延时数据的存储量非常小,由于 ΔT_i 有式(15)的约束关系成立,各个通道量化取整后的 ΔT_i 存在大量的重复,各个通道的相对延时参数 ΔT_i 存在变化时,也是按 1 递减,

并且随着聚焦深度的增加，ΔT_i 的变化速率逐渐降低，从而使得表3数据的存储变得容易实现。

根据式(10)，可以得到类似关于 T_i 与 ΔT_i 的递推关系式(16)。

$$\left.\begin{aligned} T_9 &= \Delta T_9, \\ T_8 &= \Delta T_8 + T_9, \\ &\cdots\cdots \\ T_2 &= \Delta T_2 + T_3, \\ T_1 &= \Delta T_1 + T_2。 \end{aligned}\right\} \tag{16}$$

因此，在工程实际中仅需存储表3第一排数据 ΔT_i 的初始值，并记录下1～9通道相对延时数据 ΔT_i 发生变化的位置，即可完成对表3的数据存储。在逐点聚焦的过程中，根据各个通道相对延时数据发生变化的位置，实时修正各个通道量化后的相对延时数据 ΔT_i，再根据公式(16)，即可实时计算出1～9通道的延时聚焦参数 T_i。

设 T_P，T_Q 分别是 τ_P，τ_Q 量化后的数值，根据图1、2中的对应关系，可以得到数字四波束逐点聚焦延时参数 T_P，T_Q。

$$\begin{cases} T_P = \{T_{P,4}, T_{P,8}, \cdots, T_{P,32}\} \\ \quad = \{T_4, T_6, T_8, T_9, T_7, T_5, T_3, T_1\}, \\ T_Q = \{T_{Q,4}, T_{Q,8}, \cdots, T_{Q,32}\} \\ \quad = \{T_3, T_5, T_7, T_9, T_8, T_6, T_4, T_2\}。 \end{cases} \tag{17}$$

令 $\Delta\varepsilon_i = \Delta\tau_i(F, a_i) - \gamma\Delta T_i(F, a_i)$，根据式(10)、式(16)，可以计算求出 T_i 与 τ_i 的误差 ε_i。

$$\varepsilon_i = \tau_i(F, a_i) - \gamma T_i(F, a_i) = \sum_{k=i}^{9} \Delta\varepsilon_k。 \tag{18}$$

通过 Matlab 进行数值计算求解，ε_i 的最大偏差小于5.63 ns，符合高精度逐点聚焦的要求。

2 聚焦算法的性能分析与讨论

当聚焦深度从2 mm到240 mm时，根据式(6)可知，焦点间隔为0.015 4 mm，如果直接对四波束聚焦数据取整存储，每个延时数据的存储需要2个字节，其总存储量为

$$M = 15\ 455 \times 8 \times 2 \times 2 = 494\ 560 \text{ bytes}。$$

采用笔者改进的逐点聚焦算法，仅仅需要存储表3中1～9通道的初始数值 $\Delta T_i(2\text{ mm}, a_i)(i = 1, 2, \cdots, 9)$ 和存储该数值在聚焦过程中 ΔT_i 递减1的具体位置。每个初始值的存储需要1个字节，其存储量 M_1 为

$$M_1 = (18/2) \times 1 = 9 \text{ bytes}。$$

每个 $\Delta T_i(F, a_i)$ 递减1的具体位置需要2个字节存储，其数据存储量是表1中第一排初始数值 $\Delta T_i(2\text{ mm}, a_i)$ 之和减去最后一排数值 $\Delta T_i(240\text{ mm}, a_i)$ 之和，存储量 M_2 为

$$M_2 = [\sum_{i=1}^{9}(\Delta T_i(2\text{ mm}, a_i) - \Delta T_i(240\text{ mm}, a_i))] \times 2 = (222 - 0) \times 2 = 444 \text{ bytes}。$$

因此表3的压缩存储量为 $M_1 + M_2 = 453$ bytes，存储容量仅为表1的1/1 092，可以直接利用FPGA内部的存储器来实现。关于其他不同的逐点聚焦阵元数的情况下，聚焦延时参数表的直接存储与笔者提出的压缩存储所需存储容量的对比关系如表4所示。

表4 聚焦延时参数的直接存储与压缩存储容量的对比

阵元数（中心对称）	直接存储/bytes	压缩存储/bytes	存储容量比率
8	494 560	453	1/1 092
16	989 120	1 265	1/782
32	1 978 240	3 113	1/635

如果在笔者的逐点聚焦算法中引入动态孔径技术，那么还可以进一步大幅度减小远离聚焦中心线的接收通道的延时时间，从而使得存储容量大幅度降低，同时表3的数据存储量也会大幅度降低。

笔者聚焦延时的量化因子 γ 为采样间隔 T_s 的1/8(2.5 ns/20 ns)，这样可以使得线性插值计算变成简单的移位相加运算。该逐点聚焦算法的聚焦延时参数的量化因子 γ 通常为采样间隔 T_s 的1/4，1/2，这样使得插值计算更容易。显然，聚焦延时参数的量化因子 γ 越大，$\Delta T_i(F, a_i)$ 的初值越小，$\Delta T_i(F, a_i)$ 跳变越缓慢，并且 $\Delta T_i(F, a_i)$ 的数值变化小等于1，表1数据 $\Delta T_i(F, a_i)$ 的压缩存储量越低，但是根据式(18)可知，ε_i 可能增加。

3 结 语

笔者提出了一种基于FPGA的高精度多波束逐点聚焦延时参数的压缩存储算法，其特点是将多波束聚焦的延时参数计算转化为单波束延时参数的计算，通过引入阵元间相对声程差的概念，将各个阵元接收通道的逐点聚焦延时参数转换成相邻阵元间相对延时参数的递推关系。然后采用聚焦延时量化因子 γ，将各个通道相对延时参数 $\Delta\tau_i$ 转换成单位为 γ 的整数 ΔT_i，对 ΔT_i 进行压缩存储；在聚焦过程中，通过并行加法器实时生成各个通道所需要的绝

对延时时间 T_i，根据 T_i 与聚焦波束延时参数的对应关系，将 T_i 分配给各个聚焦通道。

该逐点聚焦方法有效减少了延时聚焦参数的存储容量，同时避免了复杂的乘法运算，聚焦算法在 FPGA 中易于实现，聚焦精度可以根据聚焦延时量化因子 γ 进行动态调整。该算法适合凸阵，平阵探头等多种探头，也适合非中心对称的偏转聚焦情况。

参考文献：

[1] 陈民铀，王伟明. 基于分段动态变迹技术的超声成像方法[J]. 重庆大学学报，2010，33(3)：60-64.
CHEN Minyou，WANG Weiming. Ultrasound imaging method based on segment dynamic apodization technology [J]. Journal of Chongqing University，2010，33(3)：60-64.

[2] Camacho J，Martinez O，Parrilla M，et al. A strict-time distributed architecture for digital beamforming of ultrasound signals [J]. IEEE Transactions on Instrumentation and Measurement，2010，59(10)：2716-2723.

[3] Tamano S，Kobayashi T，Sano S Z，et al. 3D ultrasound imaging system using Fresnel ring array & high voltage multiplexer IC[C]// Proceedings of the 2004 IEEE Ultrasonics Symposium，August 23-27，2004，Montreal，Canada. Piscataway，N. J.，USA：IEEE Press，2004，1：782-785.

[4] Wall K，Lockwood G R. A new multi-beam approach to real-time 3-D imaging[C]// Proceedings of the 2002 IEEE Ultrasonics Symposium，October 8-11，Munich，Germany. Piscataway，N. J.，USA：IEEE Press，2002，2：1803-1806.

[5] Parrilla M，Brizuela J，Camacho J，et al. Dynamic focusing through arbitrary geometry interfaces[C]// Proceedings of the IEEE Ulstrasonics Symposium (IUS)，November 2-5，Beijing，China. Piscataway，N. J.，USA：IEEE Press，2008：1195-1198.

[6] Li P C. Efficient dynamic focus control for three-dimensional imaging using two dimensional arrays[J]. IEEE Transactions on Ultrasonics，Ferroelectrics and Frequency Control，2002，49(9)：1191-1202.

[7] Sohn H Y，Kang J，Cho J，et al. Time-sharing bilinear delay interpolation for ultrasound dynamic receive beamformer[J]. Electronics Letters，2011，47(2)：89-91.

[8] 沈毅，冯乃章. 医学超声成像中数字波束形成的聚焦参数压缩方法：中国，200710072422. X[P]. 2007-11-28.

[9] Aken J R V. An efficient ellipse-drawing algorithm[J]. IEEE Journal of Computer Graphics and Applications，1984，4(9)：24-35.

[10] Chang J H，Song T K. A new synthetic aperture focusing method to suppress the diffraction of ultrasound [J]. IEEE Transactions on Ultrasonics Ferroelectrics and Frequency，2011，58(2)：327-337.

[11] Bjastad T，Aase S A. Synthetic transmit beam technique in an aberrating environment [J]. IEEE Transactions on Ultrasonics，Ferroelectrics and Frequency Control，2009，56(7)：1340-1351.

[12] Song T K，Greenleaf J F. Ultrasonic dynamic focusing using an analog FIFO and asynchronous sampling[J]. IEEE Transactions on Ultrasonics，Ferroelectrics and Frequency Control，1994，41(3)：326-332.

[13] Gao C Q，Thou J Y，Koh L M，et al. Oversampled delta-sigma beamformer with modified zone-based dynamic focusing[C]// Proceedings of the 3rd IEEE International Symposium on Biomedical Imaging：Nano to Macro，April 6-9，2006，，Arlington，VA. Piscataway，N. J.，USA：IEEE Press，2006(8)：892-895.

[14] Fritsch C，Parrilla M，Sánchez T，et al. The progressive focusing correction technique for ultrasound beamforming[J]. IEEE Transactions on Ultrasonics Ferroelectrics and Frequency，2006，53(10)：1820-1831.

[15] 彭虎. 超声成像算法导论[M]. 合肥：中国科学技术大学出版社，2008.

（编辑　王维朗）