

文章编号 1004-924X(2008)07-1315-08

# 基于块位长的小波嵌入零块编码算法

王仁龙, 郝燕玲, 刘 莹

(哈尔滨工程大学 自动化学院, 黑龙江 哈尔滨 150001)

**摘要:**针对小波编码算法在编码过程需要占用大量内存的问题,提出了一种基于块位长的小波嵌入零块编码算法。该算法通过以下几项措施来减少编码过程的内存占用,同时提高编码的工作效率:(1) $LL_n$  子带单独编码;(2)提出位长定义,建立块位长位图;(3)通过位长位图代替排序,细化链表,减少内存占用;(4)通过位长位图的树结构来快速搜索重要块,克服重复扫描,加快编码速度。软件测试结果表明,对一幅  $512 \text{ pixel} \times 512 \text{ pixel} \times 8 \text{ bit}$  的图像,该算法占用的内存空间为 SPECK 算法的  $1/27$ ;硬件实现结果显示,雷达图像质量测试完全达到国际标准;该算法在保持高信噪比的同时,有效地减少了编码过程的内存占用,易于硬件实现。

**关键词:**块位长;位长位图;嵌入零块编码;内存优化

**中图分类号:** TN919.81 **文献标识码:** A

## Embedded block wavelet coding method based on block bit-length

WANG Ren-long, HAO Yan-ling, LIU Ying

(College of Automation, Harbin Engineering University, Harbin 15001, China)

**Abstract:** An embedded block wavelet coding method based on block bit-length is put forward for resolving the problem of occupying a large amount of memory in coding process for a wavelet coder. The memory occupancy can be reduced and coding efficiency can be improved through four measurements in the coding process. The four measurements mentioned above are: (1)  $LL_n$  sub-band is separately coded; (2) Block bit-length chart in terms of bit-length defined by the author is established; (3) The space of memory is saved by employing bit-length chart instead of fining lists and ranking processes; (4) The key blocks are searched by constructing bit-length chart to overcome the repeat scan and to improve the coding speed. The software experimental results show that for a piece of  $512 \text{ pixel} \times 512 \text{ pixel} \times 8 \text{ bit}$  image, occupying memory space by proposed method is  $1/27$  that of SPECK, the performance of radar image derived from hardware test can satisfy the international standard. Meanwhile, proposed method can keep the high quality of PSNR and can be implemented through hardware easily.

**Key words:** block bit-length; bit-length chart; embedded block coding; memory optimization

## 1 引言

目前典型的基于小波变换的图像编码方法有两类<sup>[1]</sup>,一类是基于小波变换所具有空频局域性特征,充分利用在不同子带内相同空间位置的小波系数之间所具有的相似性来进行编码,这一类是以 Shapiro 提出的嵌入式小波零树编码方法(Embedded Zerotree Wavelet, EZW)<sup>[2]</sup>和 Said 等提出的集合分裂树算法(Set Partitioned In Hierarchical Tree, SPIHT)<sup>[3-4]</sup>为代表的小波零树编码方法;另一类是在 SPIHT、AGP<sup>[5]</sup>、SWEET<sup>[6]</sup> 3 种编码方法基础上提出的小波嵌入零块图像编码算法(Set Partitioned Embedded Block, SPECK)<sup>[7]</sup>,SPECK 算法是基于小波变换后能量主要集中在低频子带,而不重要系数主要集中在高频子带,因而充分利用在同一子带内不重要系数的相关性来进行编码。这两类编码方法在编解码过程中都需要维护排序信息和细化信息,而用于排序信息和细化信息的链表需要占用大量的内存,既影响编码速度又不利于硬件实现。因此研究如何节省编码内存成为小波编码方法是否实用的一个关键因素。

本文提出了一种基于块位长的小波编码方法,用标志位的形式来存储各级块最大绝对值对应的位长,再通过建立的位图来快速搜索达到阈值的  $2 \times 2$  小块,从而优化内存、快速编码。软件测试及硬件实现结果表明,该算法在保持高信噪比的同时,减少了内存占用。

## 2 小波零树图像编码和小波嵌入零块图像编码

小波零树图像编码算法充分利用了不同子带内相同空间位置的小波系数之间所具有的相似性<sup>[8]</sup>,即低分辨率层的系数可以预测与它相同空间位置的高分辨率层的系数,在高阈值下不同分辨率层间的小波系数构成了零树结构;其编码思想就是将对小波变换系数的编码分解为对标志系数重要性图的编码和对重要系数的幅度编码两部分,这类小波编码都是通过排序和细化两个过程来实现的。排序过程要完成小波系数的渐进量

化,需要存储排序信息以维持排序过程,如 EZW 算法中的主列表或 SPIHT 算法中 LIS 链表存储的是这些零树结构中的不重要集合信息;细化过程要完成重要系数的位平面编码,即找出相对于量化阈值的重要系数位,需要存储重要系数以实现细化,如 EZW 算法的副链表或 SPIHT 算法中 LSP 链表就是用于存储重要系数的;所以小波零树编码算法存在占用大量内存和编码速度慢的缺点。

小波嵌入零块编码也是遵循小波零树编码思想,通过排序和细化两个过程来实现。在排序过程中,用一个 LIS 链表存储不重要集合信息,维护排序过程;在细化过程中,用一个 LSP 链表存储重要系数信息;排序和细化过程中的链表含义和功能类似于小波零树编码 EZW 算法中的主列表、副列表和 SPIHT 算法中的单向链表 LIS、LSP。不同的是,小波嵌入零块编码算法利用的是同一子带内不重要系数的相关性,在高阈值时同一子带内小波系数呈现整体不重要性,构成零块结构。同样地,小波嵌入零块编码算法也存在占用大量内存和编码速度慢的缺点。

本文在 EZW、SPIHT、SPECK 这 3 种编码方法的基础上提出了基于块位长的小波嵌入零块编码方法。与 SPECK 算法类似,利用同一子带内不重要系数的相关性来编码树状结构,不同的是该树图各节点对应的是相应块的最大绝对值对应的位长(后面简称块位长);用该树图来代替排序和细化过程中相应的链表,减少了排序和细化过程中的内存占用;采用块位长搜索策略检索树结构,当扫描某树节点时,若是该树节点没有达到阈值,则说明该树节点的所有子孙节点都没有达到该阈值,就可以不用扫描其子孙节点,直接扫描同级的其它树节点;若是该树节点达到阈值,则说明该树节点的每一级子孙节点都至少有一个节点达到该阈值,这样就可以通过树结构快速搜索出其达到阈值的块,克服了重复扫描的问题,加快了扫描速度。

## 3 小波零树图像编码和小波嵌入零块图像编码分析

### 3.1 内存优化结构

基于块位长的小波嵌入零块编码算法是通过

建立各级块位长的树状结构来进行排序、定位编码的,是以一幅图像为一整块的最大绝对值对应位长为树根,再进行 4 分裂,一直分裂到块大小为  $2 \times 2$  为止;计算出每一级的块位长作为其树节点;然后建立起位长位图。以 SPIHT 算法为例,一幅  $512 \text{ pixel} \times 512 \text{ pixel} \times 8 \text{ bit}$  的图像则需占用  $512 \text{ pixel} \times 512 \text{ pixel} \times 3 \times 18 \text{ bit}$  即 1.687 5 Mbyte 内存;而本文算法在编码过程中占用的内存为  $(1/4 + 1/16 + \dots + 1/4^n) \times 512 \text{ pixel} \times 512 \text{ pixel} \times 4 \text{ bit}$  约为 42.67 kbytes,内存耗费较少,易于硬件实现。其中块位长用 4 bit 来存储,是因为 4 bit 最大表示值为 15,那就意味着最大绝对值可以达到 15 bit,即可以表示数值达到  $2^{15} - 1 = 32\ 767$ ;而小波变换完后的小波系数绝对值完全  $< 32\ 767$ ,也就是说 4 bit 已经足够用来表示小波系数的位长了。

### 3.2 建立块位长位图

首先给出几个定义:

定义 1 位长是指小波系数绝对值取整后对应的有效位长度,不包括符号位,如式(1)所示。其中  $X_{i,j}$  为小波系数;  $|x|$  表示  $x$  的绝对值;  $\lfloor x \rfloor$  表示对  $x$  向下取整运算,例如若  $x = \lfloor 3.8 \rfloor$ ,则  $x = 3$ ;式中“=”等号用来赋值,“ $\equiv$ ”恒等号用来判断(以下同)。例:  $\text{Bit\_long}(\pm 9) = 4$ ,即  $\pm 9$  的有效位长为 4 位。

$$\text{Bit\_long}(X_{i,j}) = \begin{cases} 1 + \lfloor \log_2(|X_{i,j}|) \rfloor & |X_{i,j}| \geq 1 \\ 0 & |X_{i,j}| < 1 \end{cases} \quad (1)$$

定义 2 块位长是指在该块内小波系数最大绝对值对应的位长,如式(2)所示,其中  $T$  为该块内所有小波系数集合。

$$\text{Bit\_long}(T) = \begin{cases} 1 + \lfloor \log_2(\max_{i,j} \{|X_{i,j}| \in T\}) \rfloor & |X_{i,j}| \geq 1 \\ 0 & |X_{i,j}| < 1 \end{cases} \quad (2)$$

定义 3 位长位图的建立首先是将整幅图像分成  $(n+1)$  块,如图 1 所示,其中最低频  $LL_n$  子带独立为一块,其他的都是同一分解层 3 个子带 ( $HL_k, LH_k, HH_k$ ) 构成一块,  $k$  表示对应的分解层,  $n$  为小波分解层数;  $LL_n$  子带通过式(2)只求其块位长记为:  $\text{Bit\_T0}$ ,其他的每一块都依据 4 分裂树结构建立起相应的位长位图,设定  $HL_1, LH_1, HH_1$  这 3 个子带部分建立起相应的位长位图  $\text{Bit\_T1}$ ;类似其余几部分 ( $HL_k, LH_k, HH_k$ )

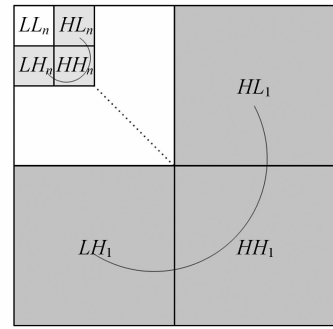


图 1 分块方式  
Fig. 1 Block division

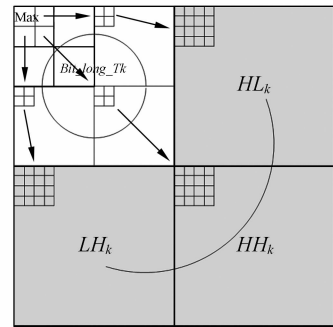


图 2 位图树结构及其与子带的关系  
Fig. 2 Tree structure and relation between structure and sub-bands

点设定为  $\text{Bit\_Tk}$ ;位长位图各级间树结构及与相应子带的关系如图 2 所示,每幅位图左上角为整幅位图的最大值。假设一幅行  $0 \sim N-1$ 、列  $0 \sim N-1$  的  $N \times N$  图像,以这种存储方式建立起位长位图  $\text{Bit\_Tk}$  是行  $0 \sim N/2^{k+1} - 1$ 、列  $0 \sim N/2^{k+1} - 1$  的  $(N/2^{k+1}) \times (N/2^{k+1})$  矩阵,位图内部的树关系除根节点只有 3 个子节点外,其他的都是 4 分裂:

$$\text{Tree}(\text{Bit\_Tk}(0,0)) = \{\text{Bit\_Tk}(0,1), \text{Bit\_Tk}(1,0), \text{Bit\_Tk}(1,1)\}.$$

$$\text{Tree}(\text{Bit\_Tk}(i,j)) = \{\text{Bit\_Tk}(2i,2j), \dots, \text{Bit\_Tk}(2i+1,2j+1) \mid 0 \leq 2i < N/2^{k+1} \text{ 且 } 0 \leq 2j < N/2^{k+1} \text{ 且 } i,j \text{ 不同时为 } 0\}.$$

位图树结构的末梢节点与其对应的  $2 \times 2$  小块内的小波系数对应关系为:

$$\text{Tree}(\text{Bit\_Tk}(i,j)) = \{X_{2i,2j}, \dots, X_{2i+1,2j+1} \mid N/2^{k+1} \leq 2i < N/2^k \text{ 或 } N/2^{k+1} \leq 2j < N/2^k\}.$$

$$\text{Tree}(\text{Bit\_T0}) = \{X_{i,j} \mid X_{i,j} \in LL_n\}.$$

根据上面给出的对应关系计算出位图树结构

各节点的值;位图树结构的末梢节点由式(3)计算,各中间子孙节点由式(4)计算,整幅位图根节

点由式(5)计算。

$$Bit\_Tk(i,j) = \begin{cases} 1 + \lfloor \log_2(\max_{X_{i,j} \in Tree(Bit\_Tk(i,j))} \{|X_{i,j}|\}) \rfloor, & \max_{X_{i,j} \in Tree(Bit\_Tk(i,j))} \{|X_{i,j}|\} \geq 1 \\ 0, & \max_{X_{i,j} \in Tree(Bit\_Tk(i,j))} \{|X_{i,j}|\} < 1 \end{cases}, \quad (3)$$

$$Bit\_Tk(i/2,j/2) = \max_{Bit\_Tk(i,j) \in Tree(Bit\_Tk(i/2,j/2))} \{Bit\_Tk(i,j)\}, \quad (4)$$

$$Bit\_Tk(0,0) = \max_{Bit\_Tk(i,j) \in Tree(Bit\_Tk(0,0))} \{Bit\_Tk(i,j)\}. \quad (5)$$

这样建立起  $n$  幅位图( $n$  为分解层数)与  $LL_n$  子带的最大绝对值位长构成了该幅图像的整体定位,而整幅图象的最大绝对值位长也就可以通过式(6)求得,为减少重复扫描奠定了基础。

$$\theta = \max_{k=0, \dots, n} \{Bit\_Tk(0,0) | n \text{ 为分解级数}; k=0 \text{ 时, 为 } Bit\_T0\}. \quad (6)$$

初始阈值定义如式(7)所示:

$$T = 2^n n = \lfloor \log_2(\max_{(i,j)} \{|X_{i,j}|\}) \rfloor, \quad (7)$$

其各级阈值为:  $T = 2^m, m = n, \dots, 0$ 。

位长与阈值的关系:由式(7)可知,当小波系数 ( $|X_{i,j}| \geq 1$ ) 为重要时,其对应那一级的阈值为  $T = 2^{Bit\_long} - 1$ 。这样在块扫描搜索时,就可以用逐级位长来扫描搜索各块,当搜索到某块重要时,再通过相应的阈值来扫描该块内小波系数。

而位图的每一级节点又对应于子带内相应的块大小及位置,这意味着可以通过这几幅位图对小波系数进行快速完全定位,从而省去排序链表 LIS;由于每块对应的位长都已经计算得出,可以清楚得知每块对应的阈值,从而省去了非重要系数表 LIP 与重要系数表 LSP。

### 3.3 基于块位长的小波嵌入零块编码算法

对一幅小波分解( $n$ 层分解)之后的图像进行基于块位长的小波嵌入零块编码步骤如下:

(1)依据 3.2 节建立位长位图,并通过式(6)选定初始位长  $\theta$ ,对应阈值为:  $T = 2^{\theta-1}$ ;

(2)利用  $n$  幅块位长位图搜索出达到各阈值的块并扫描输出;

- 对  $LL_n$  子带直接进行编码,  $CodeT(Bit\_T0)$

- 对每幅位图进行搜索;顺序为  $k = n, n-1, \dots, 1$

- \* if  $Bit\_Tk(0,0) \geq \theta$ 
  - if  $Bit\_Tk(0,0) \equiv \theta$ 

输出 1
  - SearchT( $Bit\_Tk(0,0)$ )

- \* else

输出 0

(3)令  $\theta = \theta - 1, T = 2^{\theta-1}$

- $\theta \geq 1$ , 返回步骤(2);
- $\theta < 1$ , 退出;

SearchT( $Bit\_Tk(i,j)$ )

- { 对每一个  $Bit\_Tk(x,y) \in Tree(Bit\_Tk(i,j))$

- \* if  $Bit\_Tk(x,y) \geq \theta$ 
  - if  $Bit\_Tk(x,y) \equiv \theta$ 

输出 1
  - if  $Bit\_Tk(x,y)$  有子节点,

SearchT( $Bit\_Tk(x,y)$ )

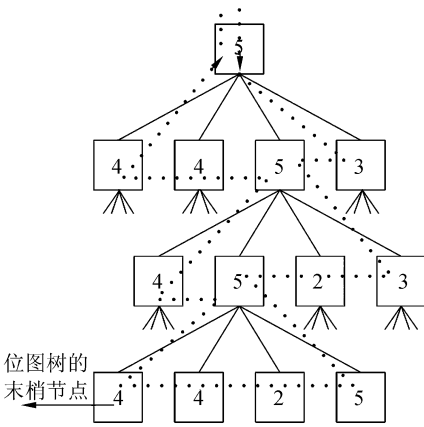


图 3 在  $\theta=5$  时一部分位图的扫描顺序

Fig. 3 Scanning order of a part of bit-length chart with  $\theta=5$

块位长搜索:由于是采用树结构来建立位图,其每个节点都对应为相应块位长。图 3 给出了树结构一部分的节点及其对应位长,还给出了该树结构一部分的搜索顺序(以虚线表示);这  $n$  幅位图都与其相应的几个分解层的子带相对应,

```

    - else
        CodeT22(Bit_Tk(x,y))

```

```

* else
    输出 0

```

```

}
CodeT22(Bit_Tk(i,j))
{ • 对每一个  $X_{i,j} \in \text{Tree}(\text{Bit\_Tk}(i,j))$  编码输出
    * if  $|X_{i,j}| \geq 2T$ 
        输出  $|X_{i,j}|$  的第  $\theta$  个最高有效值
    * else if  $T \leq |X_{i,j}| < 2T$ 
        - if  $X_{i,j} > 0$ 
            输出 10
        - else
            输出 11
    * else
        输出 0
}

```

```

CodeT(Bit_T0)
{ • 对每一个  $X_{i,j} \in \text{Tree}(\text{Bit\_T0})$  编码输出
    * if  $|X_{i,j}| \geq 2T$ 
        输出  $|X_{i,j}|$  的第  $\theta$  个最高有效值
    * else if  $T \leq |X_{i,j}| < 2T$ 
        - if  $X_{i,j} > 0$ 
            输出 10
        - else
            输出 11
    * else
        输出 0
}

```

编码一幅图像的流程简单概括如下:首先进行初始化,依据 3.2 节建立位长位图,并通过式(6)选定初始位长  $\theta$ , 对应阈值为:  $T = 2^{\theta-1}$ ; 接着搜索所有达到重要性的块,默认  $LL_n$  子带为重要块,然后编码;最后是细化工作,就是  $\theta = \theta - 1$ , 再重搜索编码。

在搜索重要块的过程中,调用函数 SearchT( $\text{Bit\_Tk}(i,j)$ )对  $\text{Tree}(\text{Bit\_Tk}(i,j))$  的子节点进行搜索;若是某子节点  $> \theta$  值,重要,但是在前面已经搜索过,就不再需要置标志位;若是某子节点等于  $\theta$  值,重要,且未搜索过,置标志位 1;若是某子节点  $< \theta$  值,不重要,置标志位 0。当某节点为重要节点时,若是该节点还有子节点,调用 SearchT()继续搜索其所有子节点;若是该节点

无子节点,说明找到重要块,调用  $\text{CodeT22}(\text{Bit\_Tk}(i,j))$  进行编码。 $LL_n$  子带为默认重要块,调用 CodeT()进行编码。

编码过程,CodeT()与 CodeT22()不同之处在于一个对应的是  $2 \times 2$  矩阵,一个对应的是  $LL_n$  子带;其编码过程是一样的。若是  $|X_{i,j}| < T$ , 不重要,输出 0;若是  $|X_{i,j}| \geq 2T$ , 重要,但是在前面已经扫描过,输出  $|X_{i,j}|$  的第  $\theta$  个最高有效值;若是  $T \leq |X_{i,j}| < 2T$ , 重要,未扫描,需要知道其标志位,再扫描;若是  $X_{i,j} > 0$ , 输出 10, 否则输出 11。

在上述算法中可知,通过建立的位图可以省去编码过程中的排序和细化链表;在保持高信噪比的同时,有效地减少了编码过程中的内存占用。同时,由于采用了块位长树结构,有效地减少了重复扫描,使得编码速度有较大提高。

## 4 软件测试

本文算法在 Pentium4 2.93 GHz 的微机上用 Visual C++ 6.0 进行验证,选用  $512 \text{ pixel} \times 512 \text{ pixel} \times 8 \text{ bit}$  的标准灰度图像 Lena 和 Barbara。小波滤波器采用 Antonini'9/7<sup>[9]</sup> 小波滤波器,采用的是提升算法<sup>[10-11]</sup>,在图像边界处采用对称延拓,熵编码算法采用自适应算术编码方案。图 4 显示了 EZW 算法、SPECK 算法、SPIHT 算法和本文算法对几幅图像的峰值信噪比曲线;表 1 为几种算法在不同码率下的峰值信噪比;表 2 给出了这 4 种算法在编码时占用的内存。图 5 列出了  $512 \text{ pixel} \times 512 \text{ pixel} \times 8 \text{ bit}$  的 Lena 和 Barbara 采用本文算法压缩后在不同码率下的重建图像。

表 1 几种算法在不同码率下的峰值信噪比

Tab.1 PSNRs for different coders at various rates

编码算法	Lena 图像(bit/pixel)			Barbara 图像(bit/pixel)		
	0.25	0.5	1.0	0.25	0.5	1.0
EZW	33.17	36.28	39.55	26.77	30.53	35.14
SPECK	34.03	37.10	40.25	27.76	31.54	36.49
SPIHT	34.11	37.21	40.44	27.58	31.40	36.41
本文算法	34.10	37.15	40.33	27.81	31.56	36.49

表 2 几种算法的内存占用表

Tab. 2 Memory occupancy for several sorts of coders

编码算法	EZW	SPECK	SPIHT	本文
占用内存	$9/2 \times N \times N$	$9/2 \times N \times N$	$27/4 \times N \times N$	$1/6 \times N \times N$

由图 4 可以看出,本文算法和 SPECK 算法、SPIHT 算法的信噪比曲线几乎重叠,这 3 种算法具有相当的信噪比,且都明显高于 EZW 算法;再分析表 1 与表 2 可知,由于本文采用的是块位长位图编码,占用的内存空间较少,对一幅  $512 \text{ pixel} \times 512 \text{ pixel} \times 8 \text{ bit}$  的图像,本文占用的内存空间为 SPECK 算法的  $1/27$ 。

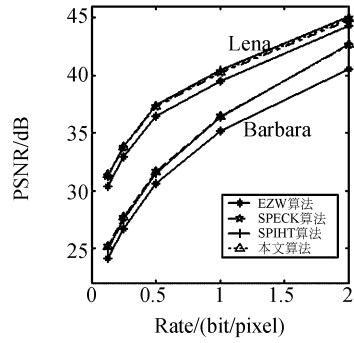
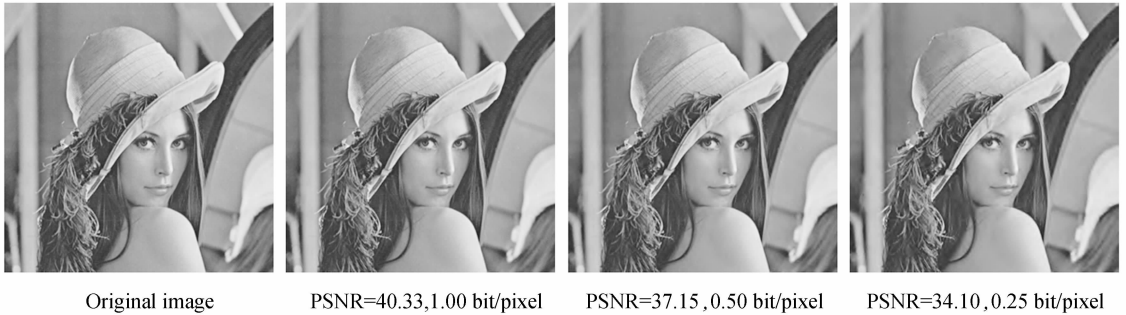


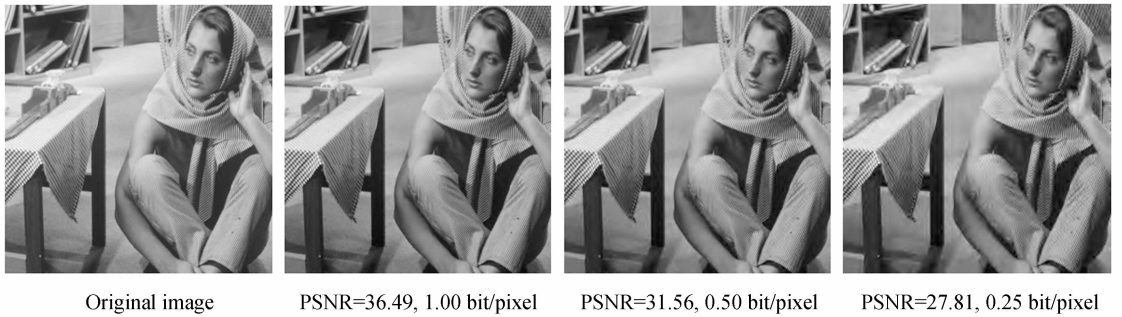
图 4 不同图像的峰值信噪比

Fig. 4 PSNRs for various images



(a) 不同码率下的 Lena 压缩图像

(a) Compressed Lena images at various rates



(b) 不同码率下的 Barbara 压缩图像

(b) Compressed Barbara images at various rates

图 5 不同码率下重建的压缩图像

Fig. 5 Compressed images reconstructed at various rates

## 5 硬件实现

### 5.1 硬件实现平台

在船舶航行数据记录仪 (Voyage Data Re-

coder, VDR) 中, 雷达图像采集压缩系统就是用于记录车载雷达主显示器屏幕上所显示的全部图像内容, 其硬件平台结构如图 6 所示。其工作原理是由 DM642 主芯片控制 A/D 芯片和 FPGA 实现雷达 VGA 格式图像的采集, 并将采集来的

数据进行压缩,最后将压缩好的数据通过网口送给主机存储。该压缩算法就是采用本文的压缩算法。

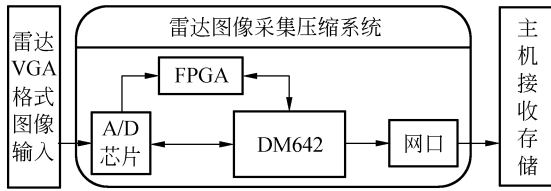


图 6 雷达图像采集压缩系统总体框图

Fig. 6 Population block diagram of radar image compressing system

## 5.2 算法移植

在移植过程中,关键是解决下面两个问题:数据的输入输出和存储空间的协调。因为基于 PC 机实现编码算法时,输入输出数据是以文件的形式存储在硬盘中的,数据的读写都由操作系统现存的文件系统管理,所以数据流的读写容易实现,并且其存储空间很大。而在 DSP 环境中,其数据都是来自 A/D 芯片,需要 DSP 及 FPGA 去控制,且由于 DSP 系统内存有限,需要协调分配。

## 5.3 性能测试

雷达图像质量测试最主要的是客观测试,采用的是国际标准测试。其测试方法是将标准提供的输入图像通过视频信号发生器输出,经本系统采集压缩后,用回放设备解压后保存为输出图像,然后将输出图像与标准输入图像逐一比较,计算出颜色误差和位置误差。

测试信号即视频输出分辨率从最低  $640 \times 350$  到最高  $1280 \times 1024$ ,帧率从 60 Hz 到 85 Hz。图 7 和图 8 分别是分辨率为  $800 \times 600$  的测试图像,在帧率为 60 Hz 时的颜色误差和位置误差测试结果,图中颜色误差门限值为 0.05,位置误差门限值为 0.002,  $TM_x$  为水平位置误差值,  $TM_y$  为垂直位置误差值。

## 参考文献:

- [1] SUDHAKAR R, KARTHIGA R, JAYARAMAN S. Image compression sing coding of wavelet coefficients; a survey[J]. *GVIP*(05), 2005, 6: 25-38.
- [2] SHAPIRO J M. Embedded image coding using zerotrees of wavelet coefficients[J]. *IEEE Transactions on Signal Processing*, 1993, 41(12): 3445-3462.
- [3] SAID A, PEARLMAN W A. A new, fast, and efficient image code based on set partitioning in hierarchical trees [J]. *IEEE Transactions on Circuits Systems, Video Technology*, 1996, 6(3): 243-250.

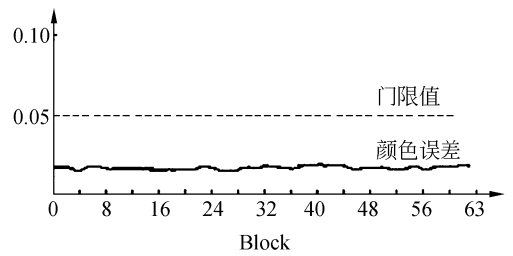


图 7 颜色误差测试结果

Fig. 7 Test result of color error

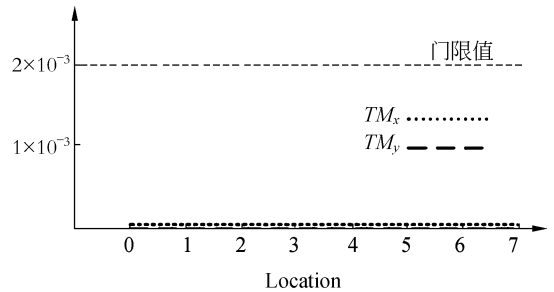


图 8 位置误差测试结果

Fig. 8 Test result of location error

## 6 结 论

针对当前的几种嵌入式小波编码算法在编码过程中都需要占用大量的内存,硬件实现难的问题,提出了一种基于块位长的小波嵌入零块编码算法。软件测试结果表明,对一幅  $512 \text{ pixel} \times 512 \text{ pixel} \times 8 \text{ bit}$  的图像,该算法占用的内存空间为 SPECK 算法的  $1/27$ 。硬件实现结果性能显示,雷达图像质量测试完全达到国际标准。该算法在保持高信噪比的同时,有效地减少了编码过程的内存占用,易于硬件实现。

- [4] 解成俊,刘艳滢. 基于提升方案与 SPIHT 算法相结合用于图像的无损压缩[J]. 光学精密工程,2002,10(6):564-568.  
XIE CH J, LIU Y Y. Research on the application of lifting scheme in image lossless compression[J]. *Opt. Precision Eng.*, 2002, 10(6):564-568. (in Chinese)
- [5] SAID A, PEARLMAN W A. Low-complexity waveform coding via alphabet and sample-set partitioning[J]. *SPIE*, 1997,3024:25-37.
- [6] ANDREW J. A simple and efficient hierarchical image coder[C]. *IEEE Int. Conf. on Image Proc.*, Washington DC, *IEEE Computer Society*, 1997:658-661.
- [7] ISLAM A, PEARLMAN W N. An embedded and efficient low-complexity hierarchical image coder [J]. *SPIE*, 1998, 3653:294-305.
- [8] 张旭东,卢国栋,冯健. 图像编码基础和小波压缩技术—原理、算法和标准[M]. 北京:清华大学出版社,2004:235-272.  
ZHANG X D, LU G D, FENG J. *Fundamentals of Image Coding and Wavelet Compressing-Principles, Algorithms and Standards*[M]. Beijing: Tsinghua University Press, 2004:235-272. (in Chinese)
- [9] ANONINI M, BARLAND M, MATHIEU P, *et al.*. Image coding using wavelet transform[J]. *IEEE Transactions on Image Processing*, 1992,2(1):205-220.
- [10] 陈升来,黄廉卿,郭静寰. 基于整型提升小波变换的图像处理及 DSP 实现[J]. 光学精密工程,2006,14(3):498-502.  
CHEN SH L, HUANG L Q, GUO J H. Image processing based on integral lifting scheme and its implementation by DSP[J]. *Opt. Precision Eng.*, 2006,14(3):498-502. (in Chinese)
- [11] 刘红,翟林培. 提升小波变换在图像压缩中的应用[J]. 光学精密工程,2005,13(增):201-205.  
LIU H, ZHAI L P. Lifting wavelet transform and its application in image compression[J]. *Opt. Precision Eng.*, 2005, 13(Supp.):201-205. (in Chinese)

作者简介:王仁龙(1980—),男,博士研究生,主要从事雷达图像采集压缩、小波分析、图像去噪等方面的研究。E-mail: wang\_renlong@163.com

郝燕玲(1944—),女,教授,博士生导师,主要从事组合导航系统方面的研究。E-mail:hyL\_407@163.com

## ●下期预告

# 基于 PSD 的地下管线探测装置设计与分析

沈林勇,李亚旻,章亚男,钱晋武

(上海大学 机电工程与自动化学院,上海 200072)

采用将 PSD(位置敏感器件, Position Sensitive Device)和其它元器件组合起来,组成介入式测量装置,通过使测量装置沿管道行进,获得管道内若干个测量点的方位,进而拟合得到地下管线的方位。适用于内径为 80~130 mm、弯曲半径为 50~1 000 m 的管线。通过实际探测的例子,得出实际测量值在水平方向上的测量误差为 7.45%,竖直方向上的测量误差位 1.07%。