

# 两种基于 DCT 变换的数字水印算法分析\*

孟兵 周良柱 万建伟 楼生强

(国防科技大学电子工程学院 长沙 410073)

**摘要** 本文介绍了多媒体数字水印的概念,对当前的技术现状进行了简单的综述,重点研究了两种 DCT 域上的数字水印算法的检测过程,在此基础上对两者的性能和特点进行了分析,并且进行了实验仿真验证。

**关键词** 版权保护,数字水印,DCT 变换,图象处理

**分类号** TN915.08

## The Analysis of Two Digital Watermark Algorithms Using DCT

Meng Bing Zhou Liangzhu Wan Jianwei Lou Shengqiang

(Department of Electronics Technology, NUDT, ChangSha, 410073)

**Abstract** In this paper, the concept of multimedia digital watermark is introduced. The present state of this technology is summarized. The hypothesis test procedures of two different algorithms using DCT are analysed, and threshold choice problem is discussed. Computer simulation is done to confirm the conclusions made.

**Key words** Copyright Protection, Digital Watermarking, DCT, Image Processing

数字技术、多媒体技术和网络技术的发展,使人们进入了数字化时代,但数字信息的普及也带来了许多新问题。由于对数字信息的拷贝非常容易,而且所得拷贝和原来的完全一样,这使得作者往往不愿把其有价值的作品放在网络上。计算机强大的数字处理能力也使得对数字信息的非法修改和使用成为可能,这成为人们进入数字化时代的一个障碍。通常的加密技术和数据校验技术并不能完全解决这一问题。加密后的数字信息只有少数人可以使用,并且一旦被解密,信息就得不到任何保护。数据验证技术可以发现信息是否被修改,但是不能满足对信息保护的要求。在这种情况下,人们提出了数字水印技术的概念,数字水印技术将特定的信息嵌入到数字信息的内容中,要求嵌入的信息有一定的稳健性,即不能被轻易的去除,在一定的条件下可以被提取出来,以确认作者的版权。

### 1 数字水印技术的分类和要求

目前数字水印技术可以分为三种:(1)稳健的不可见的水印(robust imperceptible watermark);(2)不稳健的不可见的水印(fragile imperceptible watermark);(3)可见的水印(perceptible watermark)。这三种水印技术有着不同的应用背景。第一种主要用于版权的保护,其不可见性指嵌入信息后应不引起原始信息的质量下降,这种技术也是目前研究的重点。第二种主要用于检测原始信息的改变,并能指出改变的位置和强度。第三种技术主要用于明显的标记数字信息,这往往使得原始信息的质量下降,可以用于发布免费样图等。第一种技术是最困难的,因为从信号处理的角度看,水印的稳健性和不可见性是相互矛盾的,但是,这种技术又是可能的,因为所谓的稳健性是有前提条件的——信息在处理后的质量不下降太大。对用于版权保护的数字水印有以下的基本要求:

- (1) 不可见性(imperceptible);
- (2) 安全、可靠(Secure and Reliable);
- (3) 对信号处理和攻击的稳健性(Robust)。

\* 1999年4月15日收稿  
第一作者:孟兵,男,1972年生,博士生

目前对于静态图像的数字水印算法的研究可以粗略分为如下几类:

(1) 早期的数字水印技术, 往往把不可见性放在首位, 把水印嵌入在人们对图像最不敏感的部分, 例如高频分量部分, 或图像数据的最不重要比特位(Least Significant Bits)<sup>[1]</sup>。但是, 这些方法对信号处理和有意的攻击往往不具有稳健性。

(2) 利用信号中正交变换的方法, 如基于 DCT 的方法<sup>[2][3]</sup>, 基于 DFT 的方法<sup>[4]</sup>, 基于 DWT 的方法<sup>[5][6]</sup>。

(3) 充分利用人类视觉特性的方法, 如文[7][8]。

(4) 结合当前静态图像压缩标准 JPEG 的方法, 如文[9]。

DCT 变换是一种图像处理中常用的正交变换, 其去相关的性能非常接近于 K-L 变换, 基于这种变换的 Cox 的算法<sup>[2]</sup>和 Piva 的算法<sup>[3]</sup>在当前用于静态图像的数字水印算法中是较好的。目前算法性能分析中往往忽略了对水印信息检测过程的研究, 没有给出一个比较明确的门限确定方法。这从某种意义上降低了数字水印技术的可信度, 同时使得不同算法的性能比较时无法进行定量的比较。我们认为检测门限应该由水印检测过程可以利用的信息客观地决定。本文利用概率论知识对这两种算法的检测门限的取值进行了讨论, 在此基础上对它们的性能进行了比较和分析。

## 2 Cox 算法和 Piva 算法的理论分析

### 2.1 Cox 算法分析

Cox 的算法和 Piva 算法均对原始图像嵌入一个均值为 0、方差为 1 的高斯序列, 序列  $X$  由作者的特定信息经过某种算法产生(产生方法和其安全性本文不作讨论), 嵌入的域均为图像的 DCT 变换系数。Cox 算法是首先对原始图像进行 DCT 变换, 然后对除去直流分量的 DCT 系数进行排序, 并且, 对前  $L$  个大的系数 $\{V_i, i=1 \dots L\}$  进行修改, 使得序列  $X$  嵌入到图像数据中, 最后, 进行 DCT 反变换得到加入水印信息的图像。

嵌入的公式为: 
$$V_i = V_i(1 + \alpha X_i), i = 1 \dots L \tag{1}$$

$\alpha$  用于控制嵌入信息的强度, 一般取 0.1 ~ 0.2, 检测时利用原始图像和需要检测的图像, 对(1)式进行相反的计算得到另一个序列, 将它进行线性变换得到一个零均值, 方差为 1 的序列  $X'$ 。通过(2)式计算  $X$  与  $X'$  的相关性, 检验被检图像是否为利用序列  $X$  进行过水印嵌入处理的图像。

$$\text{sim}(X, X') = \frac{1}{L} \sum_{i=1}^L X_i X'_i / L \tag{2}$$

由于序列  $X$  是利用作者的特定信息产生, 其他人很难得到与  $X$  相关性大的序列, 故其他人利用伪造的

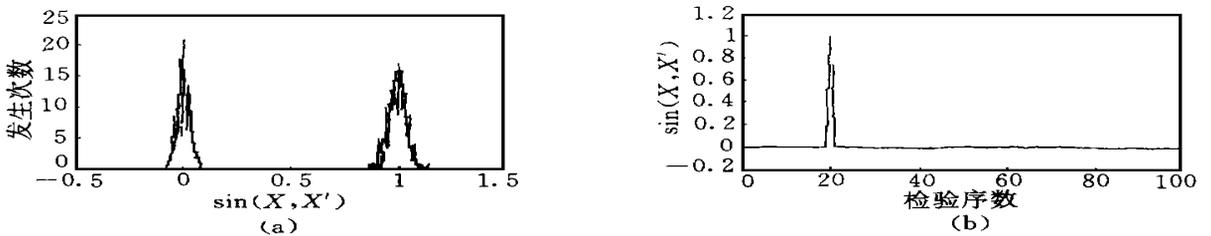


图 1 Cox 算法中检验统计量的分布和检测性能  
Fig. 1 The distribution of the test variable and the detection performance

序列经过(2)式, 只能得到接近 0 的结果。对于被检验的图像 P, 假设检验为  $H_0$ : P 没有嵌入序列,  $H_1$ : P 嵌入了序列。在  $H_0$  假设下,  $X$  与  $X'$  不相关, 统计量  $\text{sim}(X, X')$  仍然为正态分布,  $\text{sim}(X, X') \sim N(0, 1/L)$ 。在  $H_1$  假设下统计量  $L \cdot \text{sim}(X, X')$  服从  $L$  个自由度的开方分布,  $L \cdot \text{sim}(X, X') \sim \chi^2(L)$ 。

图 1(a) 是在两种假设条件下对检验统计量  $L \cdot \text{sim}(X, X')$  的分布的实验结果, 实验利用 Lena (256x256) 图像分别在  $H_0$  和  $H_1$  假设条件下 500 次统计了各个  $\text{sim}(X, X')$  值的发生次数。如果经

过序列  $X$  嵌入处理的图像, 在相应的检测下可以得到显著检测结果。图 1.(b) 是对 100 幅嵌入不同序列的 Lena(256x256) 图像进行检测的结果(只有第 20 幅嵌入的序列与检测使用的  $X$  是一致的)。该方法所嵌入的信息的位置是原始图像的最重要区域, 故对各种处理也有很强的稳健性。一般而言, 嵌入水印的图像经过各种处理之后, 检验只能基于  $H_0$  假设, 因为  $\text{Sim}(X, X)$  在  $H_0$  假设下服从正态分布, 可以利用通常的正态分布随机变量检测方法, 将门限设在  $3 \sim 5\sigma$  处( $\sigma^2$  为  $H_0$  假设条件下方差的估计值), 取  $3\sigma$  时的漏检概率小于  $10^{-3}$ 。

### 2.2 Piva 算法分析

Piva 算法是另一种 DCT 域上的数字水印嵌入方法。它首先对原始图像进行 DCT 变换, 然后对变换的系数进行 ZigZag 扫描转换, 选取从第  $L$  个数开始的  $M$  个连续的中频区域系数  $T = \{t_{L+1}, \dots, t_{L+M}\}$ , 通过(3)式进行信息嵌入, 其中  $\alpha$  取值在 0.1 到 0.2 之间。最后将改变后的系数插入到 DCT 系数矩阵中做 DCT 反变换得到

$$t_{L+i} = t_{L+i} + \alpha t_{L+i} x_i \tag{3}$$

加入水印后的结果。检测过程通过(4)式的检测统计量进行,  $t_i^*$  为被检验图像的相应 DCT 系数。原文将

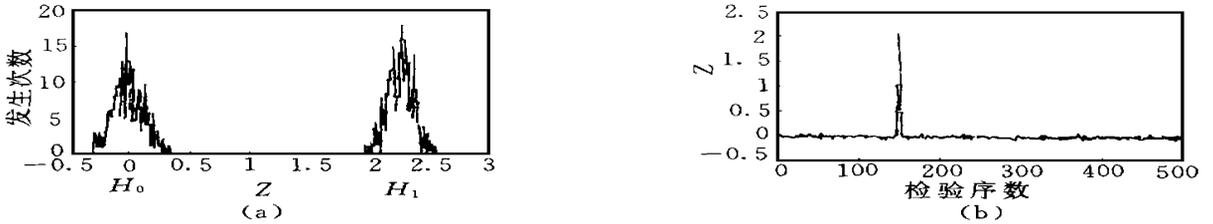


图 2 Piva 算法中检验统计量的分布和检测性能

Fig. 2 The distribution of the test variable of Piva algorithm and the detection performance

$t_i^*$  看成随机变量, 进行了公式的推导, 这里将  $t_i^*$  作为非随机的参数进行讨论。具体的推导见本文附录。

$$z = \frac{1}{M} \sum_{i=1}^M x_i t_{L+i}^* \tag{4}$$

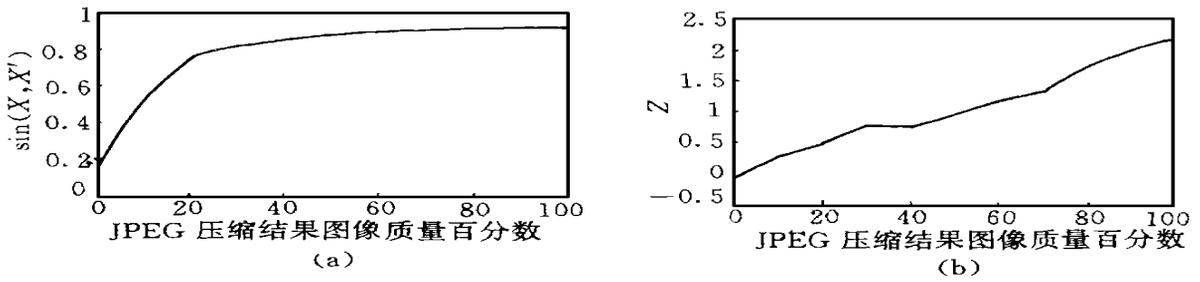
对被检验图像 P, 可建立假设检验的两个命题,  $H_0$ : P 没有嵌入序列  $X$ ,  $H_1$ : P 嵌入了序列  $X$ 。在  $X_i \sim N(0, 1)$ , 以及  $\forall i \neq j$  时  $X_i, X_j$  不相关的条件下, 通过推导可以得到在不同假设下的概率分布, 在  $H_1$  假设下,  $z \sim N(\alpha \sum_{i=1}^M t_{L+i} / M, (1 + 2\alpha^2) \sum_{i=1}^M t_{L+i}^2 / M^2)$ 。在  $H_0$  假设下如果没有经过水印嵌入处理  $z \sim N(0,$

$\sum_{i=1}^M t_{L+i}^2 / M^2)$ , 如果嵌入了和  $X$  不同的序列则  $z \sim N(0, (1 + \alpha^2) \sum_{i=1}^M t_{L+i}^2 / M^2)$ 。由于  $\alpha$  的值很小, 可见在  $H_0$

和  $H_1$  假设下,  $z$  的方差近似相等, 并且可以由被检验图像的 DCT 系数估计, 不同假设下  $z$  的概率分布可由图 2(a) 的实验结果说明, 实验利用 Lena(256x256) 图像分别在  $H_0$  和  $H_1$  假设条件下, 500 次统计各个  $z$  值的发生次数(实验中  $\alpha = 0.2, L = 10000, M = 16000$ )。图 2(b) 是对 500 幅嵌入不同序列  $X$  的 Lena(256x256) 图像进行检测的结果(只有第 150 幅嵌入的序列与检测使用的  $X$  是一致的)。由于嵌入水印的图像经过各种处理后, 将不再满足前面的假设  $H_1$ , 我们仍然可以利用  $H_0$  进行假设检验的过程。可以利用  $t_i^*$  对  $z$  的方差  $\sigma^2$  进行计算, 将检测的门限定在  $3 \sim 5\sigma$  处进行检测。

### 3 Cox 算法和 Piva 算法对信号处理的稳健性实验结果

由于自然图像的 DCT 系数在 ZigZag 扫描的方向上的包络是衰减的, 故 Cox 算法所修改的 DCT 系数往往是低频部分, 在图像质量下降不多的条件下, 对带有低通滤波特性的处理而言, 这是图像中最为稳定的部分, 所以 Cox 算法是目前数字水印技术中性能非常出色的。图 3(a) 是用 Cox 算法嵌入水印的图像经过 JPEG 压缩处理后的检测结果, 横坐标为图像质量的百分数, 纵坐标为相关值。而 Piva 算法, 嵌入的位置是在图像的中频区域, 对这类处理的稳健性要差于第一种方法, 图 3(b) 为实验结果。这往往也是检测时利用原始图像方法与检测时不利用原始图像方法相比共有的特点。但是, 对于加入随机噪声的处理来说, 其往往干扰整个频带, 而且噪声的频谱一般有白色的特性。Cox 算法由于嵌入信息的

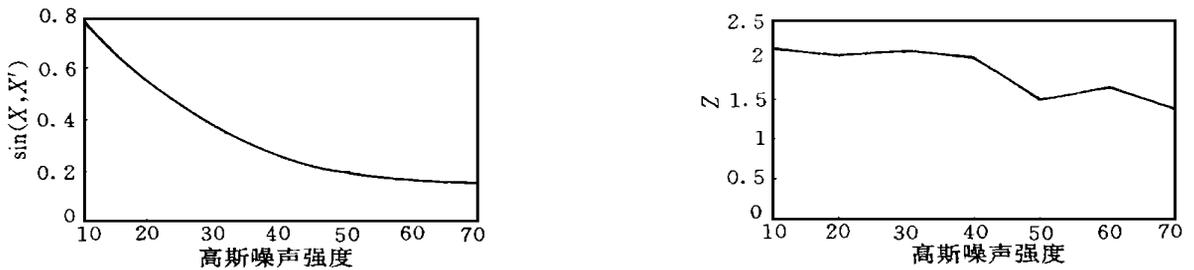


图中的  $x'$  为检测门限 ( $3\sigma$ )

图3 两种算法对于 JPEG 压缩处理的结果

Fig. 3 The result of two algorithms' performance to JPEG compression

位置在低频区域, 为保证图像质量, 嵌入序列的长度比 Piva 算法要短, 检测时必须利用公式(1) 求出序列  $X$ , 而  $X$  受干扰的影响较大。Piva 算法由于是利用公式(4) 直接计算统计量进行检测, 嵌入序列的长度比 Cox 算法长, 该检测过程类似于扩频通信中的抗干扰方法, 其性能应该比 Cox 算法要好。图4(a) 和图4(b) 的实验结果使上述分析结论得到了验证, 实验利用 PhotoShop 5.02 对 Cox 算法和 Piva 算法的结果加入 Gauss 噪声, 横坐标为 Gauss 噪声的强度参数, 可见 Piva 算法对这种处理有较好的稳健性。对于其他的带有低通滤波特性的处理方法如中值滤波等, Cox 算法的效果较好, 而对于低频分量影响较大的处理如直方图均衡等, Piva 算法显示了较强的稳健性, 表 1 的实验结果验证了这些结论。



(a) 图中的  $x'$  为检测门限 ( $3\sigma$ ) (b)

图4 两种算法对加入高斯噪声干扰的结果

Fig. 4 The result of two algorithms' performance to Gaussian noise interference

表 1 两种算法对于四种常用图像处理的结果

Tab. 1 The results of the two algorithms to 4 general image process methods

		直方图均衡	对比度调整	Gamma 校正	中值滤波
Cox 算法:	$\text{Sim}(X, X')$	0.4744	0.3820	0.3740	0.7998
	$3\sigma$	0.2466	0.2949	0.2013	0.0985
Piva 算法:	$Z$	3.2722	2.7937	2.1325	0.4228
	$3\sigma$	0.5294	0.5090	0.3518	0.2338

#### 4 结合两种算法的实验结果和结论

通过以上分析和实验, 发现对于不同的处理方法, Cox 算法, 和 Piva 算法各有优势。对于 Cox 算法的检测需要利用原始图像, 这在许多情况下是不方便的。可以利用两者的特点, 用 Cox 算法和 Piva 算法同时对图像嵌入水印(为了保持水印的不可见性嵌入的强度需要调整), 这样经过水印嵌入的图像就具有了两者的优点, 两者嵌入的区域在频域上是相互分离的, 不存在相互干扰的问题。检测时可以先利用 Piva 算法进行无须原始图像的检验, 可以解决相当大部分的检测处理。当检测出现无法确认的情况时, 可以借助原始图像, 进行 Cox 算法的检测, 这样既提高了水印算法的稳健性, 又具有使用时的灵活性。图 5(a) 为原始的 Lena (256x256) 图像; 图 5(b) 是利用两种算法同时嵌入水印的结果(Piva 算法的嵌入

强度  $\alpha$  取 0.1, 其他参数不变); 图 5(c) 为图 5(b) 和图 5(a) 的差(为便于观察进行了亮度的调整); 图 5(d) 为嵌入水印后加入强度为 50 的 Gauss 噪声的被检测图像; 图 5(e) 为嵌入水印后经过 JPEG 压缩后 (10% 的图像质量) 的被检测图像。两种水印的嵌入算法, 在 DCT 系数域上是分离的, 所以检测的性能和前面的分析是一致的。



图 5 同时利用两种算法嵌入信息的结果

Fig. 5 The results of using the two algorithms to embed information

以上的实验结果和对两种算法的理论分析是一致的, 可见本文讨论的检测门限的选取方法是合理的。这样可以定量地对不同的算法进行性能的比较和分析。对于其他的算法, 也可以利用概率论和数理统计的方法进行类似的处理。实验结果亦说明, 对于频域上的数字水印算法嵌入信息应该包含各个频段, 以增强对信号处理的稳健性。

本文分析了静态图象数字水印中的两种算法, 特别是水印检测中的门限确定问题。通过实验发现两种算法对于不同的信号处理方法各有优势, 结合两者同时进行信息的嵌入是可行的, 并且对此进行了实验。对两种算法同时进行检测的理论分析, 以及如何充分利用 HVS 的特性自适应的确定嵌入信息的强度需要进一步的研究。

### 参考文献

- 1 Schyndel. A Digital Watermark. IEEE Int. Conf. Image Processing, 1994
- 2 Cox. Secure Spread Spectrum Watermarking for Multimedia. IEEE Trans. Image Processing, Dec 1997.
- 3 Piva. DCT-based Watermark Recovering without Resorting to Uncorrupted Original Image. IEEE Int. Conf. Image Processing, 1997
- 4 Ruanaidh. Phase watermarking of images. IEEE Int. Conf. Image Processing, 1996.
- 5 Deepa Kunder. A robust watermarking method using wavelet-based fusion. IEEE Int. Conf. Image Processing, 1997
- 6 Deepa Kunder. Digital Watermarking Using Multiresolution Wavelet Decomposition. IEEE INT. Conf. ASSP 1998
- 7 Podilchuk and Zeng. Digital Watermarking using Visual Model. SPIE Int. Conf on Human Vision and Electronic Image, 1997
- 8 Swanson. Transparent robust image watermarking. IEEE Int. Conf. Image Processing, 1996
- 9 Koch and Zhao. Towards robust and hidden image copyright labeling. IEEE Workshop on Nonlinear Signal and Image Processing, 1996

### 附录

Piva 算法中检验统计量为  $z = \sum_{i=1}^M x_{it_{L+i}} / M$ , 下面对不同的情况给出它的概率分布, 给出  $z$  在不同条件下的均值和方差。

1. 待检测图像没有利用 Piva 算法进行任何信息的嵌入。

此时  $t_{L+i} = t_{L+i}$ ,  $E(z) = E(\sum_{i=1}^M x_{it_{L+i}}) / M = 0$ , 即  $z \sim N(0, \sum_{i=1}^M t_{L+i}^2 / M^2)$ 。

2. 待检测图像利用 Piva 算法嵌入了序列  $X$  (与检测时用的  $X$  相同)。

此时检验统计量为  $z = \sum_{i=1}^M (x_{it_{L+i}} + \alpha t_{L+i} x_i^2) / M$ , 对于特定的图象而言  $t_i$  是非随机的参数, 式中的第一项为  $\{x_i\}$  的线性组合, 仍然为正态分布, 第二项为自由度为  $M$  的开方分布。因为  $M$  很大, 由开方分布的性质, 第二项也近似为正态分布。

$$E(z) = \sum_{i=1}^M (E(x_{it_{L+i}}) + E(\alpha t_{L+i} x_i^2)) / M = \sum_{i=1}^M \alpha t_{L+i} / M。$$

$$E(z^2) = E(\sum_{i=1}^M \sum_{j=1}^M (x_{it_{L+i}} + \alpha t_{L+i} x_i^2)(x_{jt_{L+j}} + \alpha t_{L+j} x_j^2)) / M^2 = \sum_{i=1}^M \sum_{j=1}^M E(x_{it_{L+i}} x_{jt_{L+j}}) / M^2 + \sum_{i=1}^M \sum_{j=1}^M E(x_{it_{L+i}} \alpha t_{L+j} x_j^2) / M^2 + \sum_{i=1}^M \sum_{j=1}^M E(\alpha t_{L+i} t_{L+j} x_i^2 x_j^2) / M^2$$

$$= \sum_{i=1}^M t_{L+i}^2 / M^2 + \sum_{i=1}^M \sum_{j=1, j < i}^M \alpha^2 t_{L+i} t_{L+j} / M^2 + \sum_{i=1}^M 3\alpha^2 t_{L+i}^2 / M^2 \quad (\text{下转第 92 页})$$

and Simulation, January 11-14, 1998, San Diego, CA, USA

2 Giuseppe Lazeolla and Andrea D'Ambrgio. A Web-Based Environment for the Reuse of Simulation Models. 1998 International Conference on Web-based Modeling and Simulation, January 11-14, 1998, San Diego, CA, USA

3 The Silk Homepage. Introduction to Silk. 1997, <http://www.threadtec.com>

4 Object Management Group. The Common Object Request Broker: Architecture and Specification (Revision 2.0). July 1995

(上接第79页)

$$E^2(z) = \prod_{i=1}^M \prod_{j=1}^M \alpha^2 t_{L+i} t_{L+j} / M^2$$

$$D(z) = E(z^2) - E^2(z) = (1 + 2\alpha^2) \prod_{i=1}^M t_{L+i}^2 / M^2$$

$$\text{故 } z \sim N\left(\prod_{i=1}^M \alpha t_{L+i} / M, (1 + 2\alpha^2) \prod_{i=1}^M t_{L+i}^2 / M^2\right).$$

3. 待检验图像利用 Piva 算法嵌入了序列  $X$  ( $X$  与检测时用的  $X$  不同)。

此时 
$$z = \prod_{i=1}^M x_{iL+i} / M = \prod_{i=1}^M x_i(t_{L+i} + \alpha t_{L+i} x_i) / M, \text{ 其中 } \{x\} \text{ 与 } \{x\} \text{ 不相关。}$$

$$E(z) = 0$$

$$E(z^2) = E\left(\prod_{i=1}^M \prod_{j=1}^M x_i t_{L+i} x_j t_{L+j} / M^2\right) + E\left(\prod_{i=1}^M \prod_{j=1}^M \alpha x_i t_{L+i} t_{L+j} x_j\right) / M^2$$

$$+ E\left(\prod_{i=1}^M \prod_{j=1}^M \alpha x_j t_{L+j} t_{L+i} x_i\right) / M^2$$

$$+ E\left(\prod_{i=1}^M \prod_{j=1}^M \alpha^2 t_{L+i} t_{L+j} x_i x_j\right) / M^2 = (1 + \alpha^2) \prod_{i=1}^M t_{L+i}^2 / M^2$$

$$\text{故 } z \sim N\left(0, (1 + \alpha^2) \prod_{i=1}^M t_{L+i}^2 / M^2\right)$$