

基于 LMS 算法的图片降噪

黄博文

摘要：本文使用 LMS 算法进行图片降噪，并对该方法进行改进，检验改进后方法的效果，并与基于 DnCNN 网络方法所得的降噪效果进行比较。并且改进后的方法效果接近于 DnCNN 方法。

关键字：LMS；图片降噪

Abstract: In this paper, LMS algorithm is used for image noise reduction, and the method is improved, the effect of the improved method is tested, and the noise reduction effect obtained based on the DnCNN network method is compared. And the effect of the improved method is close to the DnCNN method.

Key words: LMS, Image Denoising

引言

图像系统中光电管噪声、摄像管噪声、前置放大器噪声、光学噪声等是影响图片噪声的主要几种噪声来源，这些噪声加入图片后会导致图片模糊不清，影响观感，因此需要进行图片降噪处理，空域像素特征去噪算法、变换域去噪算法、BM3D 去噪算法等都是传统的去噪算法，其中 BM3D 算法效果最好，新兴的去噪算法如基于 DnCNN 网络的去噪算法经过适当的训练也是效果不错。本文基于 LMS（least mean square）算法对图片进行去噪处理。

1. LMS 自适应滤波

自适应信息处理在通讯、多维自适应处理、雷达信号处理中的自适应波束形成、武器系统中的自适应跟踪及自适应制导、遥感技术中的自适应滤波、图像影音资料的自适应修复等领域^[1]有着极大的作用。

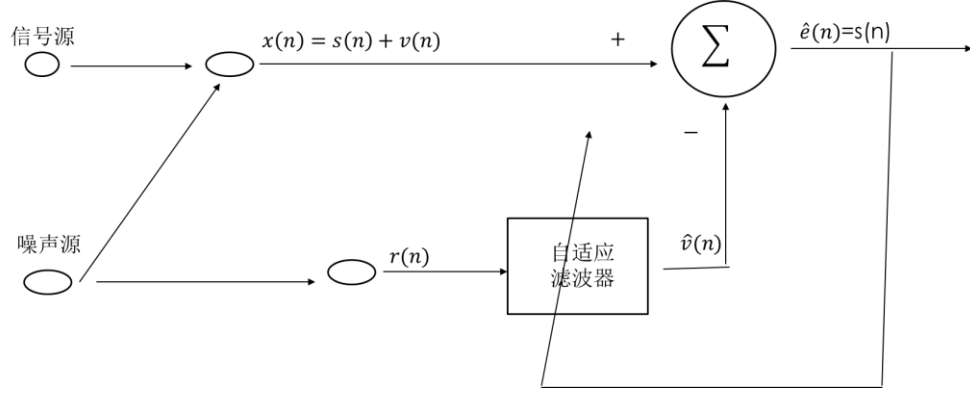


图 1.1 自适应噪声抵消原理方框图

如图 1.1 所示为自适应噪声抵消原理示意图，

$x(n)$ 为主信号；

$s(n)$ 为有用信号；

$v(n)$ 为噪声；

$r(n)$ 为参考信号，与 $v(n)$ 或 $s(n)$ 相关；

$\hat{v}(n)$ 是 $v(n)$ 的最佳估计。

为了从主信号中获得有用信号，首先对噪声进行估计，

$$\begin{aligned}\hat{v}(n) &= \sum_{m=0}^M w_m r(n-m) \\ &= w_0(n)r(n) + w_1(n)r(n-1) + \cdots + w_M(n)r(n-M), 0 \leq m \leq M\end{aligned}\quad (1.1)$$

其中 M 表示滤波器的阶数， $r(n-m)$ 由 $r(n)$ 延时获得。

具有 M 个权重滤波器的估计误差 $e(n)$ 由下式定义：

$$e(n) = x(n) - \hat{v}(n) = x(n) - w^T(n)r(n) \quad (1.2)$$

其中

$$w(n) = \begin{bmatrix} w_0(n) \\ w_1(n) \\ w_2(n) \\ \vdots \\ w_M(n) \end{bmatrix}$$

$$r(n) = \begin{bmatrix} r(n) \\ r(n-1) \\ r(n-2) \\ \vdots \\ r(n-M) \end{bmatrix}$$

对式 (1.2) 取平方和数学期望得

$$e(n)^2 = x(n)^2 - 2x(n)r(n)^T w(n) + w(n)^T r(n)r(n)^T w(n) \quad (1.3)$$

$$E(e(n)^2) = E(x(n)^2) - 2E(x(n)r(n)^T)w(n) + w(n)^T E(r(n)r(n)^T)w(n) \quad (1.4)$$

记

$$\begin{aligned} P &= E(x(n)r(n)^T) \\ R &= E(r(n)r(n)^T) \end{aligned}$$

则式 (1.4) 简化为

$$E[e(n)^2] = E[x(n)^2] - 2P^T w + w(n)^T R w(n) \quad (1.5)$$

式 (1.5) 是一个关于 w 权矢量的二次函数, 由于 R 为 $r(n)$ 的自相关矩阵, $r(n) \neq 0$ 因此 R 严格正定且满足 Toeplitz 对称性, 若将 $E[e(n)^2]$ 作为代价函数, 则必存在极小点, 这一点上一阶梯度为零。

$$\frac{\partial E[e(n)^2]}{\partial w} = -2P + 2Rw \quad (1.6)$$

令式 (1.6) 等于零得

$$w = R^{-1}P \quad (1.7)$$

实际上 P 和 R 的统计量是未知的, Widrow 和 Hoff 提出了用迭代的方法求解式 (1.7) 的方法。如果参考信号 $r(n)$ 和噪声 $v(n)$ 相关, 则自适应滤波器将在输出端去除其相关性, 具体做法是: 从参考信道的噪声中产生一个主信道中背景噪声的估计值 $\hat{v}(n)$, 然后从主信道中减去这个估计噪声, 那么自适应噪声滤波器的输出就是有用信号 $\hat{s}(n)$ 。

用最速下降法得到式 (1.7) 的解

自适应滤波器被更新的第 $n+1$ 步迭代式为

$$w(n+1) = w(n) + \Delta w(n) \quad (1.8)$$

将 $\Delta w(n)$ 用 $-\mu \frac{\partial \varepsilon(w(n))}{\partial w}$ 代替 (μ 为收敛参数)

$$w(n+1) = w(n) - \mu \frac{\partial \varepsilon(w(n))}{\partial w} \quad (1.9)$$

将 $\frac{\partial \varepsilon(w(n))}{\partial w}$ 用丢弃期望算子的瞬时梯度代替

$$\frac{\partial \varepsilon(w(n))}{\partial w} = 2E \left[e(n) \frac{\partial e(n)}{\partial w} \right] = -2E[e(n)r(n)] = -2e(n)r(n) \quad (1.10)$$

结合式 (1.9), 滤波器权重被更新为

$$w(n+1) = w(n) + 2\mu e(n)r(n) \quad (1.11)$$

综上所述, 基于最小均方误差算法的自适应噪声抵消算法可按如下步骤实现:

第一步, 设一个初值 $w_m(0)$;

第二步, 由式 (1.1) 计算自适应滤波器的输出 $\hat{v}(n)$;

第三步, 估计当前时刻 n 的误差 $e(n)$;

第四步, 用最速下降法更新滤波器权重 w ;

第五步, 校验误差是否满足标准, 若满足, 则停止迭代, 若不满足, 进行下一步;

第六步, $n \rightarrow n+1$, 到下一时刻, 重复以上步骤, 直到满足要求停止。

收敛参数 μ 的选择,

$$0 < \mu < \frac{1}{R}$$
$$0 < \mu < \frac{1}{\lambda_{max}}$$

λ_{max} 为自相关矩阵 R 的最大特征值。由于并不知道 R ，因此，实际情况下， μ 总是试探性的选取，既保证收敛速度，又保证收敛。

2. 图片降噪

如下图 2.1 所示为一张风景照片，图 2.2 为加入均值为 0，方差为 0.01 的高斯噪声后的照片。现使用前面介绍的 LMS 算法对图 2.2 进行降噪。



图 2.1 原图

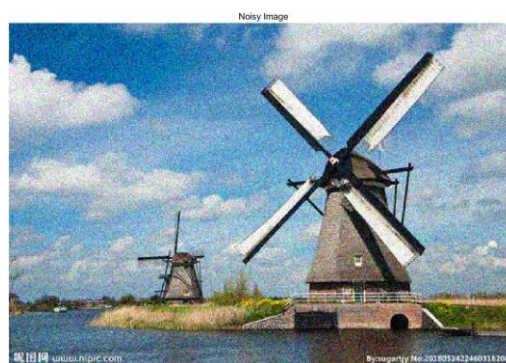


图 2.2 含噪图片 ($\mu = 0, \sigma^2 = 0.01$)

2.1 像素及图片的 RGB 格式处理

图 2.2 为一个像素为 334×499 的照片，在 RGB 格式下，每一个像素点对应相应的 RGB 色彩值^[2] (如图 2.3 所示)。则图 2.1 可由一个 $334 \times 499 \times 3$ 的三维数组表示(如图 2.4 所示)。

























顏色	英文名稱	RGB	16色	顏色	英文名稱	RGB	16色
	Snow	255 250 250	#FFFAFA		PaleTurquoise1	187 255 255	#BBFFFF
	GhostWhite	248 248 255	#F8F8FF		PaleTurquoise2	174 238 238	#AEEEEE
	WhiteSmoke	245 245 245	#F5F5F5		PaleTurquoise3	150 205 205	#96CDCD
	Gainsboro	220 220 220	#DCDCDC		PaleTurquoise4	102 139 139	#66888B
	FloralWhite	255 250 240	#FFFAF0		CadetBlue1	152 245 255	#98F5FF
	OldLace	253 245 230	#FDF5E6		CadetBlue2	142 229 238	#8EE5EE
	Linen	250 240 230	#FAF0E6		CadetBlue3	122 197 205	#7AC5CD
	AntiqueWhite	250 235 215	#FAEBD7		CadetBlue4	83 134 139	#53868B
	PapayaWhip	255 239 213	#FFEFD5		Turquoise1	0 245 255	#00F5FF
	BlanchedAlmond	255 235 205	#FFEBCD		Turquoise2	0 229 238	#00E5EE
	Bisque	255 228 196	#FFE4C4		Turquoise3	0 197 205	#00C5CD
	PeachPuff	255 218 185	#FFDAB9		Turquoise4	0 134 139	#00868B

图 2.3 RGB 配色表（部分）

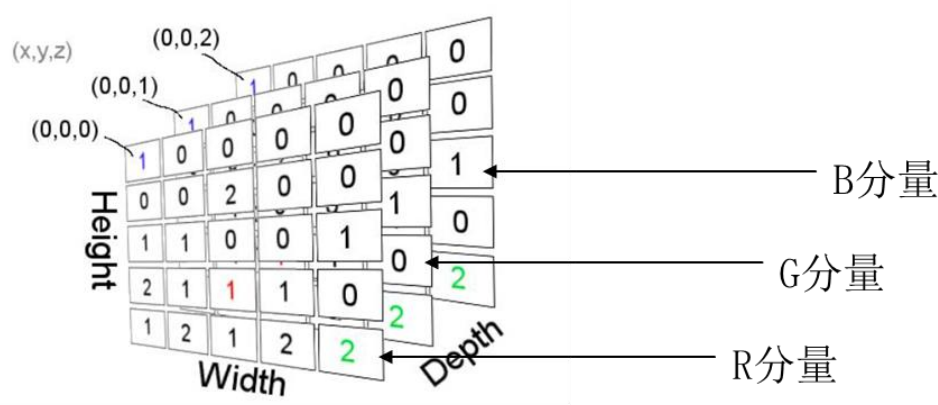


图 2.4 图片的 RGB 格式三维数组示意图

现在只讨论其中一个分量——R 分量，

$$\begin{bmatrix}
 a_{11} & a_{12} & a_{13} & \cdots & \cdots & \cdots & a_{1,497} & a_{1,498} & a_{1,499} \\
 a_{21} & a_{22} & a_{23} & \cdots & \cdots & \cdots & a_{2,497} & a_{2,498} & a_{2,499} \\
 a_{31} & a_{32} & a_{33} & \cdots & \cdots & \cdots & a_{3,497} & a_{3,498} & a_{3,499} \\
 \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\
 \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\
 \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\
 a_{332,1} & a_{332,2} & a_{332,3} & \cdots & \cdots & \cdots & a_{332,497} & a_{332,498} & a_{332,499} \\
 a_{333,1} & a_{333,2} & a_{333,3} & \cdots & \cdots & \cdots & a_{333,497} & a_{333,498} & a_{333,499} \\
 a_{334,1} & a_{334,2} & a_{334,3} & \cdots & \cdots & \cdots & a_{334,497} & a_{334,498} & a_{334,499}
 \end{bmatrix}$$

该分量可写成分块矩阵的形式，

$$[A_1 \ A_2 \ A_3 \ \cdots \ \cdots \ \cdots \ A_{497} \ A_{498} \ A_{499}]$$

再将其按如下方式排列成一个 166666 维的列向量，

$$\begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ \vdots \\ \vdots \\ \vdots \\ A_{497} \\ A_{498} \\ A_{499} \end{bmatrix} \quad (2.1)$$

记 (2.1) 式为 \check{R} ，G、B 分量做相同操作，得 \check{G} 、 \check{B} 。

对图 2.2 的 \check{R} 、 \check{G} 、 \check{B} 进行滤波得图 2.5，图 2.6，图 2.7 所示效果。

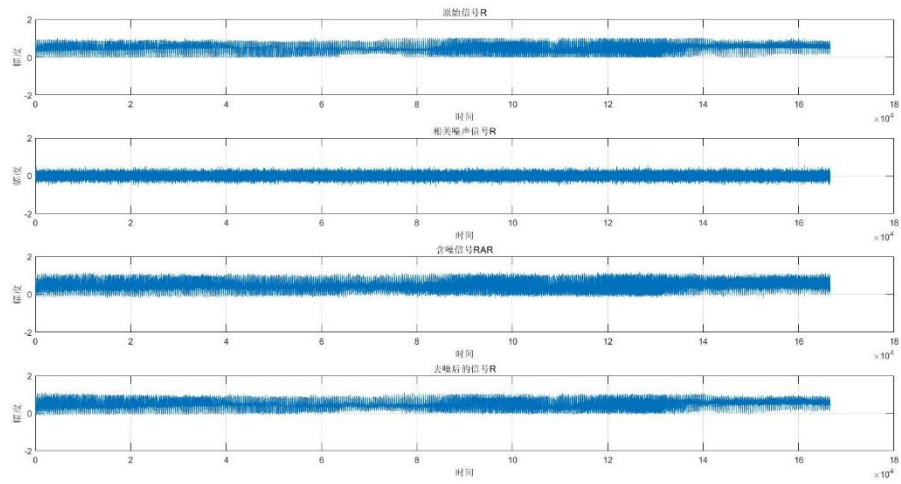


图 2.5 R 分量滤波效果图

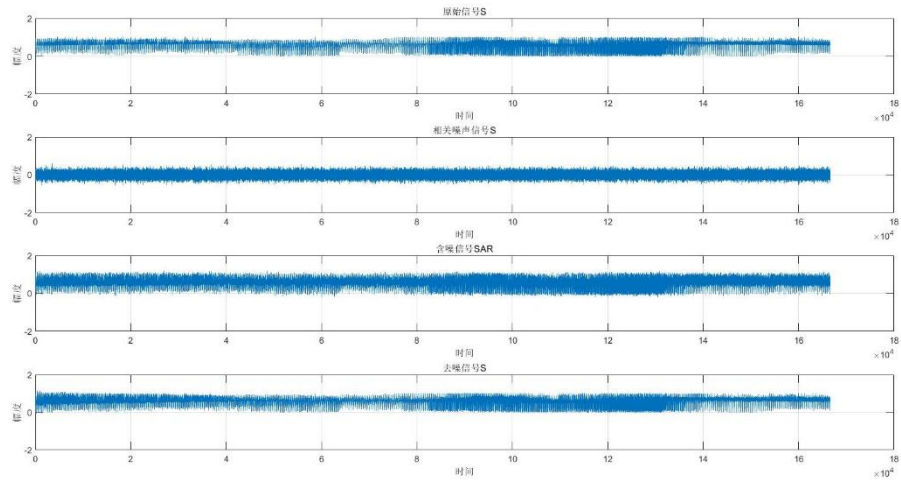


图 2.6 G 分量滤波效果图

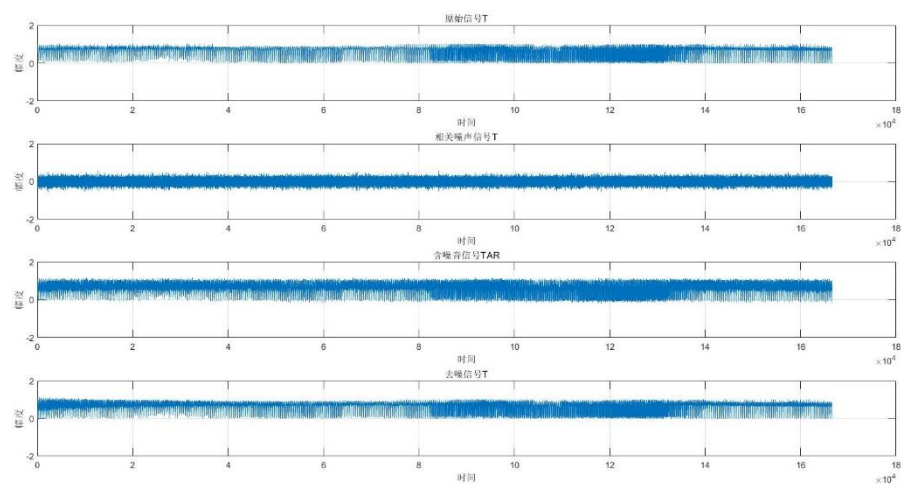


图 2.7 B 分量滤波效果图

定义剩余噪声为原图与滤波之后图片对应分量之差,并画出序号为 115000 到 115500 步的原信号与滤波后信号及剩余噪声图,可见,在后半部分,剩余噪声较小,而前半部分剩余噪声较大。后半部分原信号与滤波后信号误差较小。

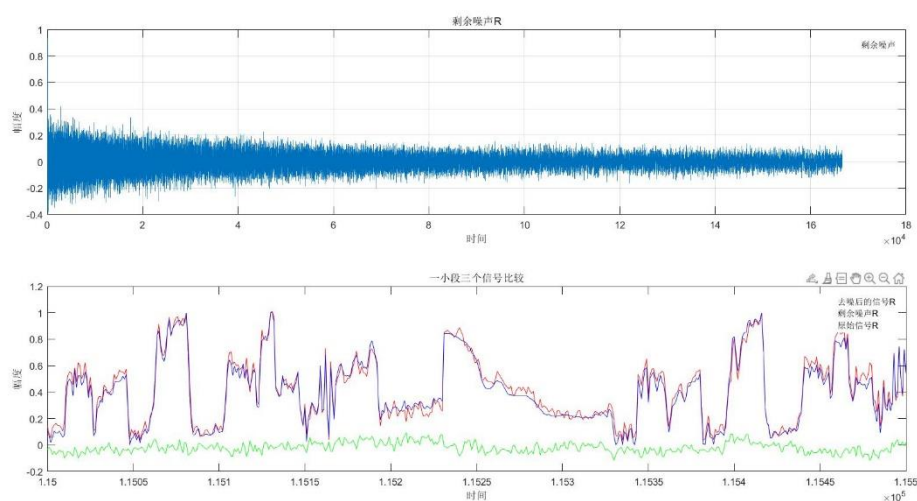


图 2.8 R 分量剩余噪声及小段三信号比较图

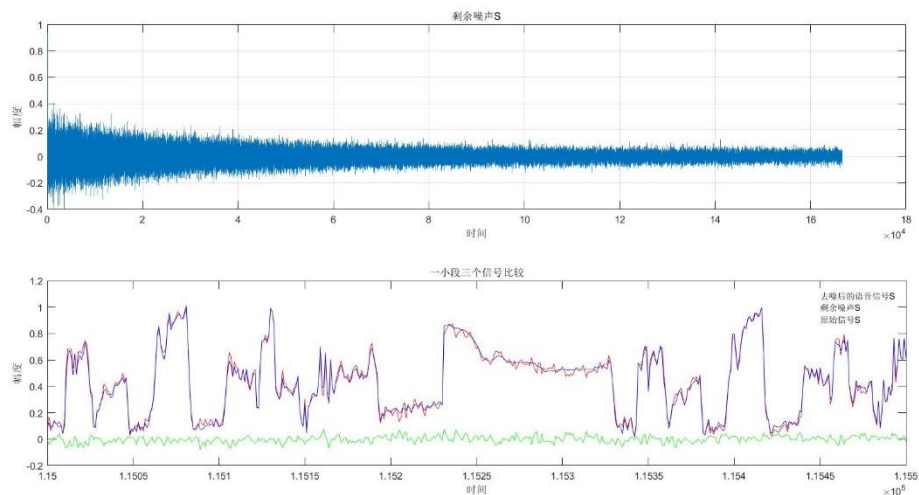


图 2.9 G 分量剩余噪声及小段三信号比较图

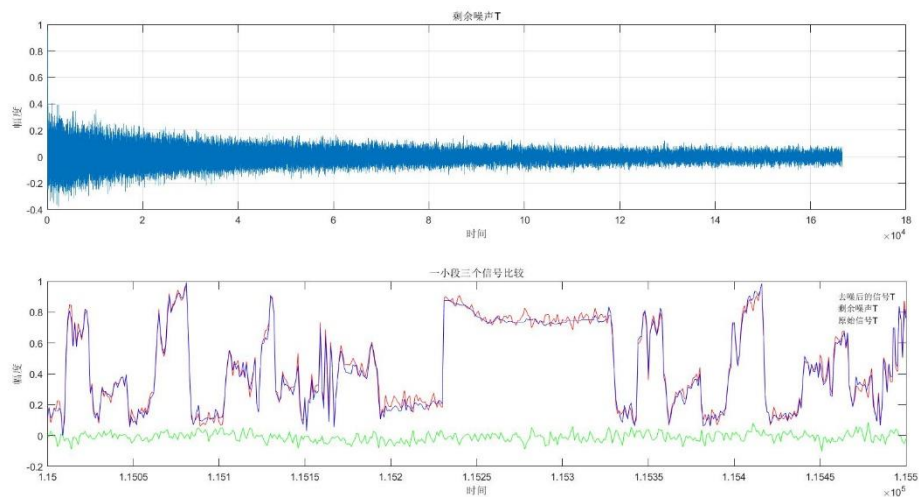


图 2.10 B 分量剩余噪声及小段三信号比较图

将滤波后的各分量合并后，生成滤波后的图片如图 2.11 所示。

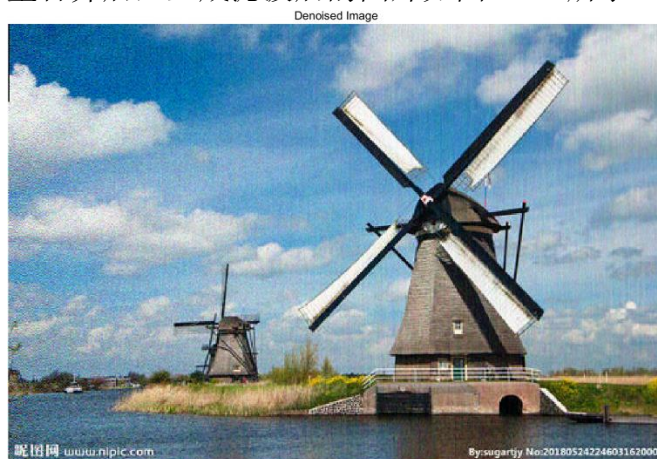


图 2.11 滤波后图片

从图 2.11 可以看出，图片右边的滤波效果优于左边，且越靠近左边效果越差，左上角出现了一段黑色条带，丢失了一部分信息。结合图 2.8、图 2.9、图

2.10 知左端剩余噪声过大，因此需要做以下处理：

如图 2.12 所示，先计算 R 分量滤波后的结果 \check{R} ，再将 R 分量翻转后滤波得 \check{R}_1 ，将 \check{R} 及 \check{R}_1 的前半段截去，后半段合并。剩余噪声减小。

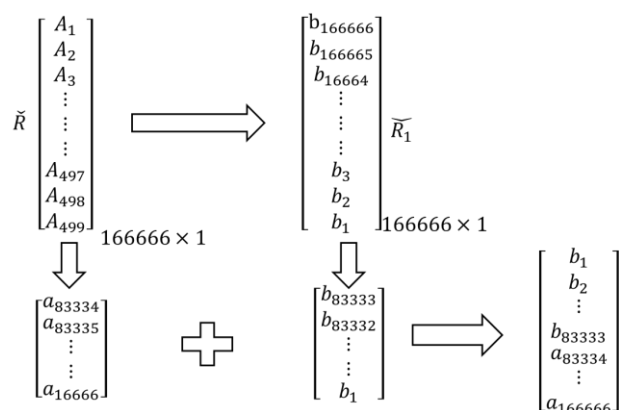


图 2.12 剩余噪声阶段处理图

经过处理后，三个分量的噪声如图 2.13 所示，

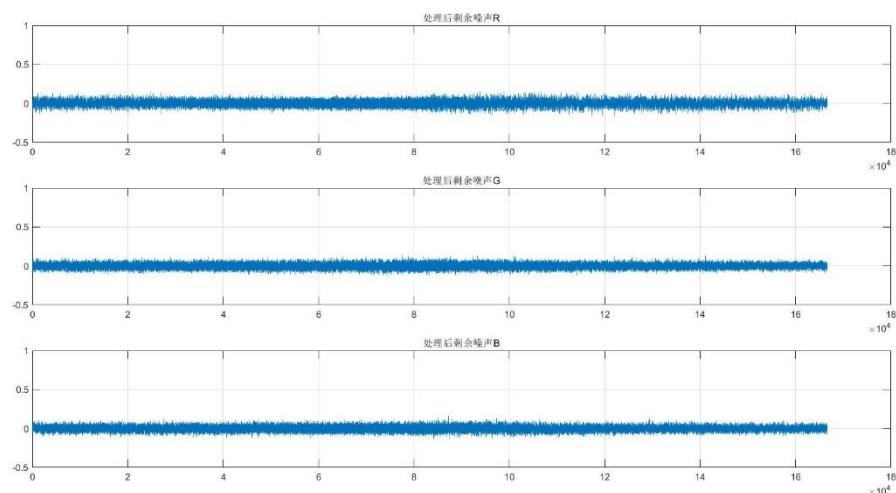


图 2.13 R、G、B 分量剩余噪声图

将三个分量合并，得降噪后得图片



图 2.14 截取部分剩余噪声后的图片

2.2 与 DnCNN 网络处理效果进行对比

根据 Kai Zhang 等人^[3]于 2017 年提出的 DnCNN 网络,使用 MATLAB 自带的实时脚本文件 RemoveGaussianNoiseFromColorImageUsingCNNExample.mlx,对图 2.2 所示图片进行处理,得图 2.15 所示效果。



图 2.15 DnCNN 网络处理效果图

观察图 2.14 及图 2.15 知图 2.15 更明亮清晰,而图 2.14 似乎存在一些暗点,但是比较两幅图中的文字发现,图 2.14 更好的保留了文字信息,而图 2.15 中的问题显得模糊不清。另外,由于人眼对亮度等的敏感程度不同^[4],肉眼观测去评判两种方法的效果有一定的不合理性,因此,引入图像处理常用的两个量来评估其效果,分别为峰值信噪比(PSNR)和结构相似度(SSIM)。

表 2.1 LMS 与 Dncnn 方法 PSNR 及 SSIM 值比较表

方法	PSNR (dB)	SSIM
噪声图像	20.2099	0.6376
LMS	26.1656	0.8670
改进 LMS	30.2861	0.9254
DnCNN	29.2864	0.9370

由表 2.1 知,含噪声的图像峰值信噪比和结构相似度都较低,使用 LMS 方法进行滤波后,这两个量都有所增大,改进后,峰值信噪比达到了 30.2861,高于 DnCNN 方法的 29.2864,大约高出了 1dB,结构相似度为 0.9254,相较于 DnCNN 方法,低出 0.0116。尽管两种方法在视觉上存在一定的差异,但是使用 PSNR 及 SSIM 度量得两种方法效果相近,因此,可见改进后得 LMS 方法取得了不错的效果。

3. 总结和展望

本文使用 LMS 算法及改进的 LMS 算法对图片进行了降噪处理,并与 DnCNN

方法处理后的结果进行了对比。并且认为改进后的 LMS 算法取得了不错的效果。

本文只处理了高斯噪声,并未对椒盐噪声、泊松噪声、乘性噪声等进行处理,因此改进后的 LMS 方法对这些噪声的处理效果有待进一步研究,另外,本文计算次数有限,需要多次重复计算,观察对于不同方差的高斯噪声的效果;此外,对于图像颜色的表示不止一种格式,还可以基于 HSV 格式进行滤波,在此工作中,已对使用 DnCNN 方法的 HSV 格式进行了计算,发现其效果不如 RGB 格式,但是改进后 LMS 算法,未进行计算。

参考文献:

- [1] 邹国良, 自适应滤波理论及其应用[M], 河北大学出版社, 1997, 04
- [2] RGB 配色表[DB/OL], <http://www.wahart.com.hk/rgb.htm>
- [3] Zhang Kai, Zuo Wangmeng, Chen Yunjin, Meng Deyu, Zhang Lei. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. [J]. IEEE transactions on image processing : a publication of the IEEE Signal Processing Society, 2017, 26(7).
- [4] 贾永红, 数字图像处理(第三版), 武汉大学出版社, 2015, 7
- [5] 潘士先, 谱估计和自适应滤波[M], 北京航空航天大学出版社, 1991. 07