

# 一种有效的脉冲噪声滤波算法

周锐锐, 毕笃彦, 毛柏鑫

(空军工程大学 工程学院, 陕西 西安 710038)

**摘要:**提出了一种新的图象脉冲噪声消除算法,该方法可以快速有效地去除图象中的脉冲噪声,并保留图象的细节。该算法主要通过利用象素点邻域的中值与最大、最小值信息,决定噪声象素的取值。对本算法与中值滤波算法进行了分析仿真,结果表明,本算法去除噪声与保留细节的效果好,对高强度的脉冲噪声也有比较好的滤波效果。

**关键词:**滤波;脉冲噪声;去噪

**中图分类号:**TP391 **文献标识码:**A **文章编号:**1009-3516(2004)06-0044-03

一般情况下,图象常被噪声所污染。抑制噪声,提高信噪比,一直是图象预处理的重要问题。一些常见的噪声有椒盐(Salt & Pepper)噪声、脉冲噪声、高斯噪声<sup>[1]</sup>等。脉冲噪声含有随机的白强度值(正脉冲噪声)或黑强度值(负脉冲值),线形滤波器去除高斯噪声的效果很好,但对椒盐噪声、脉冲噪声效果一般,它的主要问题是有可能模糊图象中的尖锐、不连续部分,为克服这一缺点,提出了非线性滤波器<sup>[2]</sup>。基于统计排序的中值滤波(MF)是当前应用最广泛的非线性抑噪方法之一,滤除脉冲噪声的效果较好。但是中值滤波对所有象素采用统一处理方法,既滤除了噪声,也改变了真正信号点的值,造成了图象模糊。近年来,出现了多种基于中值滤波的改进算法,例如加权中值滤波<sup>[3]</sup>、自适应中值滤波<sup>[4]</sup>、中心加权滤波<sup>[5]</sup>、开关中值滤波<sup>[6]</sup>、伪中值滤波<sup>[7]</sup>等,这些算法在改善中值滤波器的性能方面做出了有益的探索,滤除脉冲噪声的效果有了很大的改进,但这些算法在滤波之前,需要准确判断、标识出噪点。基于Min-max算法<sup>[8]</sup>的噪声峰值滤波器在细节模糊问题上得到了一定的解决,但当噪声密度大时,滤波效果很差。而且该算法需要迭代多次,计算速度较慢。针对这些问题,本文提出了一种快速、有效的脉冲噪声消除算法,在滤除噪声的同时,保留了较多的细节信息,没有造成图象的模糊,而且在噪声密度较大时也可以取得较好的效果。

## 1 算法的实现及原理分析

脉冲噪声随机地改变图象中某点的灰度值,在图象中表现为黑色、白色的亮斑,在数值上表现为很大或很小。由经验可知,对于自然图象,相邻点之间存在着很大的相关性,某一点的灰度值与其周围点的灰度值非常接近,除了孤立点(一般认为是噪声)外,即使在边缘部分也是这样。在一幅图象中,如果一个象素点的值远大于或小于其邻域的值,也就是说,该象素点与其邻域的相关性很小,那么该点很可能已被噪声污染了,需要修复;否则,如果其值与邻域点值很接近,就应该是一个有效的信号点<sup>[9]</sup>。基于上述考虑,我们给出判别区间 $[T_1, T_2]$ ,在一幅图象 $[v(i, j)]$ 中,如果一点 $(i, j)$ 的灰度值 $v(i, j)$ 超出判别区间,就进行修正,反之,保持原值不变,即: $v(i, j) < T_1, v(i, j) > T_2$ 时, $v'(i, j) = v(i, j)$ ;  $T_1 < v(i, j) < T_2$ 时 $v(i, j) = v(i, j)$ 。 $v'(i, j)$ 表示经过滤波后的点 $(i, j)$ 的灰度值。

文献[10]提出了一种峰值谷值(peak-and-valley)滤波算法,该方法是对图象中的点进行窗口操作,在一个窗口内,将除中心点外其它点排序,若中心点值大于最大值,峰值就用最大值代替该点值;若中心点值小

收稿日期:2004-05-23

作者简介:周锐锐(1978-),女,陕西宝鸡人,博士生,主要从事图象处理,数据压缩,数据融合等研究;  
毕笃彦(1962-),男,陕西扶风人,教授,博士生导师,主要从事图象处理,数据压缩,模式识别等研究;  
毛柏鑫(1939-),男,辽宁法库人,教授,主要从事图象处理,数据压缩,模式识别等研究。

于最小值, 谷值就用最小值代替该点值。这种方法考虑到了相邻点之间的相关性, 但当噪声密度大时效果很差, 这是因为, 噪声密度大时, 在一个窗内包含有不只一个噪点, 排序后, 是用噪声值对中心点值进行了修复, 使得噪声无法被滤除。针对上述问题, 本文提出了一种新的滤波算法, 下面给出具体步骤:

1) 对图象  $[v(i, j)]$  中的点  $v_{ij}$  进行窗口操作  $W[v(i, j)]$ , 滤波窗口为  $m \times n(m, n)$  (为不小于 3 的奇整数)。除点  $v_{ij}$  外的窗口内的点记为  $\{v_1, v_2, \dots, v_N\}, N = m \times n - 1$ , 将  $v_1, v_2, \dots, v_N$  排序:  $v_{(1)} \leq v_{(2)} \leq \dots \leq v_{(N)}$ 。然后取其中值  $V_{med} = \text{med}(W[v(i, j)])$ , 最大值  $V_{max} = \max(W[v(i, j)])$ , 最小值  $V_{min} = \min(W[v(i, j)])$ 。

2) 确定阈值  $T_1, T_2$ 。  $T_1 = V_{med} - (V_{med} - V_{min})\alpha; T_2 = V_{med} + (V_{max} - V_{med})\alpha$ , 其中  $0 < \alpha < 1$ ,  $\alpha$  的取值可随噪声密度的增大而适当减小, 通过实验分析, 一般取 0.9 左右。

3) 将窗口  $W$  的中心点  $v_{ij}$  的灰度值  $v(i, j)$  与阈值  $T_1, T_2$  进行比较, 若该点的灰度值  $v(i, j) \notin [T_1, T_2]$  就进行修正, 反之, 保持原值不变。即:  $v(i, j) \leq T_1$  时,  $v'(i, j) = T_1; v(i, j) \geq T_2$  时,  $v'(i, j) = T_2; T_1 < v(i, j) < T_2$  时,  $r'(i, j) = r(i, j)$ 。

## 2 实验与结果分析

本文以一幅  $256 \times 256$  的标准图象为例, 用中值滤波(分别采用  $3 \times 3$  和  $5 \times 5$  的窗口)和本文算法(采用  $3 \times 3$  的窗口,  $\alpha = 0.9$ )进行了仿真实验。每次实验加入密度不同的椒盐噪声(正脉冲噪声和负脉冲噪声各占 50%), 以输出图象的信噪比为标准评价滤波器的性能。如果仍由  $[v(i, j)]$  表示输入图象,  $[v'(i, j)]$  表示输出图象, 而  $[s(i, j)]$  表示未加噪声的原始图象, 则待处理输入及输出图象信噪比分别为

$$R_{SN, in} = 10 \lg \left( \frac{\sum_{i,j} s^2(i, j)}{\sum_{i,j} [s(i, j) - v(i, j)]^2} \right); R_{SN, out} = 10 \lg \left( \frac{\sum_{i,j} s^2(i, j)}{\sum_{i,j} [s(i, j) - v'(i, j)]^2} \right)$$

在表 1 中, 相对于输入图象的噪声密度, 分别给出了输入待处理图象, 经窗口  $3 \times 3$  和  $5 \times 5$  的 MF 滤波和本文算法滤波的信噪比。

表 1 不同噪声密度情况下的各滤波器输出图象信噪比 dB

噪声密度/p	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
输入 SNR	16.13	7.11	5.31	4.08	3.12	2.35	1.67	1.09	0.60
MF3 × 3SNR	15.09	14.53	13.17	11.03	8.65	6.40	4.29	2.61	1.28
MF5 × 5SNR	14.63	14.52	14.27	13.89	12.95	10.82	7.67	4.56	2.07
本文算法 SNR	18.58	16.77	15.21	14.09	13.01	11.83	9.92	7.30	3.83

图 1~9 给出了不同噪声密度情况下用 MF 滤波和用本文算法滤波的视觉效果。从图中可以直观地看出, 中值滤波的去噪效果依赖滤波窗口的大小。窗口小, 可以保留图象的细节部分, 但去噪效果差; 窗口大去噪效果较好, 但损失太多的图象细节, 造成图象模糊。而本文提出的算法在各种输入噪声密度下, 输出信噪比都明显大于 MF, 滤波速度快, 效果好。



图 1 原始图象



图 2 加入 20% 椒盐噪声

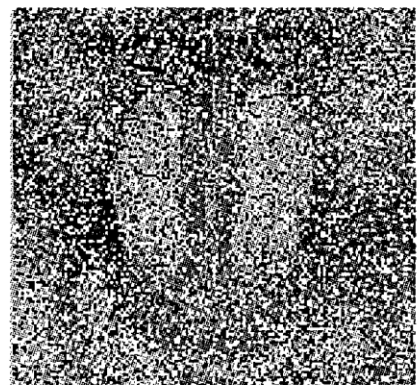


图 3 加入 40% 椒盐噪声

于最小值,谷值就用最小值代替该点值。这种方法考虑到了相邻点之间的相关性,但当噪声密度大时效果很差,这是因为,噪声密度大时,在一个窗内包含不止一个噪点,排序后,是用噪声值对中心点值进行了修复,使得噪声无法被滤除。针对上述问题,本文提出了一种新的滤波算法,下面给出具体步骤:

1)对图象 $[v(i,j)]$ 中的点 $v_{ij}$ 进行窗口操作 $W[v(i,j)]$ ,滤波窗口为 $m \times n(m,n)$ (为不小于3的奇整数)。除点 $v_{ij}$ 外的窗口内的点记为 $\{v_1, v_2, \dots, v_N\}$ , $N = m \times n - 1$ ,将 $v_1, v_2, \dots, v_N$ 排序: $v_{(1)} \leq v_{(2)} \leq \dots \leq v_{(N)}$ 。然后取其中值 $V_{med} = \text{med}(W[v(i,j)])$ ,最大值 $V_{max} = \max(W[v(i,j)])$ ,最小值 $V_{min} = \min(W[v(i,j)])$ 。

2)确定阈值 $T_1, T_2$ 。 $T_1 = V_{med} - (V_{med} - V_{min})\alpha$ ;  $T_2 = V_{med} + (V_{max} - V_{med})\alpha$ ,其中 $0 < \alpha < 1$ , $\alpha$ 的取值可随噪声密度的增大而适当减小,通过实验分析,一般取0.9左右。

3)将窗口 $W$ 的中心点 $v_{ij}$ 的灰度值 $v(i,j)$ 与阈值 $T_1, T_2$ 进行比较,若该点的灰度值 $v(i,j) \notin [T_1, T_2]$ 就进行修正,反之,保持原值不变。即: $v(i,j) \leq T_1$ 时, $v'(i,j) = T_1$ ;  $v(i,j) \geq T_2$ 时, $v'(i,j) = T_2$ ;  $T_1 < v(i,j) < T_2$ 时, $r'(i,j) = r(i,j)$ 。

## 2 实验与结果分析

本文以一幅 $256 \times 256$ 的标准图象为例,用中值滤波(分别采用 $3 \times 3$ 和 $5 \times 5$ 的窗口)和本文算法(采用 $3 \times 3$ 的窗口, $\alpha = 0.9$ )进行了仿真实验。每次实验加入密度不同的椒盐噪声(正脉冲噪声和负脉冲噪声各占50%),以输出图象的信噪比为标准评价滤波器的性能。如果仍由 $[v(i,j)]$ 表示输入图象, $[v'(i,j)]$ 表示输出图象,而 $[s(i,j)]$ 表示未加噪声的原始图象,则待处理输入及输出图象信噪比分别为

$$R_{SN,in} = 10 \lg \left( \frac{\sum_{i,j} s^2(i,j)}{\sum_{i,j} [s(i,j) - v(i,j)]^2} \right); R_{SN,out} = 10 \lg \left( \frac{\sum_{i,j} s^2(i,j)}{\sum_{i,j} [s(i,j) - v'(i,j)]^2} \right)$$

在表1中,相对于输入图象的噪声密度,分别给出了输入待处理图象,经窗口 $3 \times 3$ 和 $5 \times 5$ 的MF滤波和本文算法滤波的信噪比。

噪声密度/p	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
输入 SNR	16.13	7.11	5.31	4.08	3.12	2.35	1.67	1.09	0.60
MF3 × 3SNR	15.09	14.53	13.17	11.03	8.65	6.40	4.29	2.61	1.28
MF5 × 5SNR	14.63	14.52	14.27	13.89	12.95	10.82	7.67	4.56	2.07
本文算法 SNR	18.58	16.77	15.21	14.09	13.01	11.83	9.92	7.30	3.83

图1~9给出了不同噪声密度情况下用MF滤波和用本文算法滤波的视觉效果。从图中可以直观地看出,中值滤波的去噪效果依赖滤波窗口的大小。窗口小,可以保留图象的细节部分,但去噪效果差;窗口大去噪效果较好,但损失太多的图象细节,造成图象模糊。而本文提出的算法在各种输入噪声密度下,输出信噪比都明显大于MF,滤波速度快,效果好。

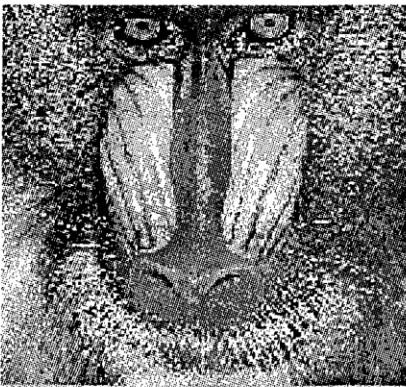


图1 原始图象

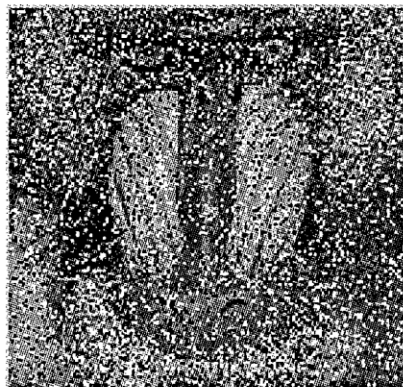


图2 加入20%椒盐噪声

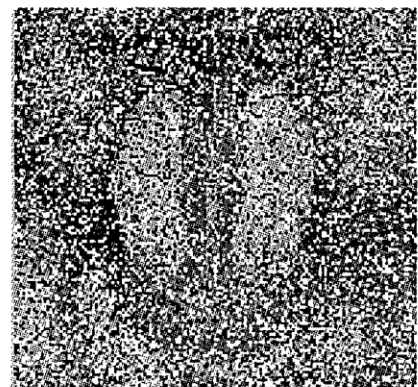


图3 加入40%椒盐噪声

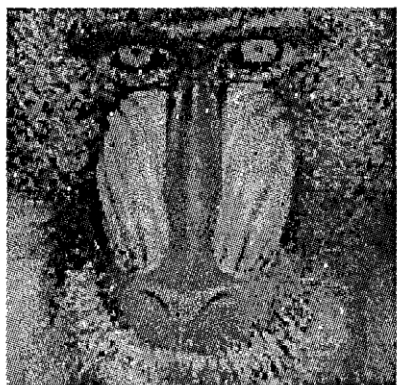


图4 20% 椒盐噪声  
3×3MF 滤波



图5 20% 椒盐噪声  
5×5MF 滤波

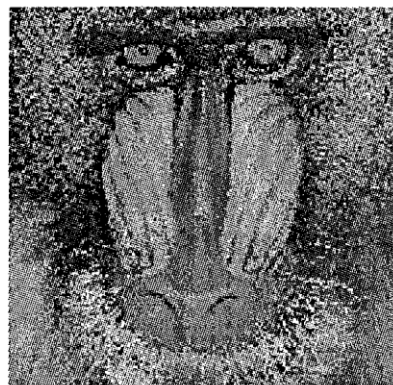


图6 20% 椒盐噪声  
本文算法 3×3MF 滤波



图7 40% 椒盐噪声  
3×3MF 滤波

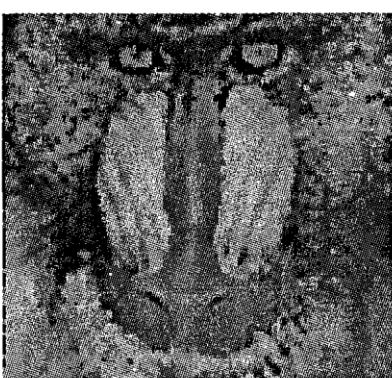


图8 40% 椒盐噪声  
5×5MF 滤波

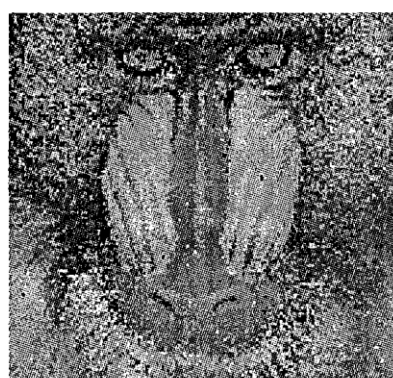


图9 40% 椒盐噪声  
本文算法 3×3MF 滤波

### 3 结束语

本文针对一般非线性滤波中噪声滤除与图象细节的保留之间的矛盾,提出了一种简单有效的滤波算法。本算法充分考虑了相邻像素之间的相关性,适当选择了判别区间,而不是简单地用一个窗内的中值、最大值或最小值来修复噪点。实验结果表明,在有效地去除图象中的脉冲噪声的同时,能够保留较多的图形细节,在噪声密度高的情况下,也取得了较好的效果。

#### 参考文献:

- [1] 丁国栋,向新,张敬利. 高斯白噪声背景下的自扩信号检测[J]. 空军工程大学学报(自然科学版),2003,2(4):59-61.
- [2] 贾云德. 机器视觉[M]. 北京:科学出版社,2002.
- [3] Brownrigg D. The weighted median filter[J]. Commun Assoc Comput,1984,27(8):807-818.
- [4] Restrepo A, Bovik A C. Adaptive trimmed mean filter for image restoration[J]. IEEE Trans Acoust, Speech, signal Processing,1998,36(8):1326-1337.
- [5] Ko S J, Neuvo Y. Center weighted median filter and their applications to image enhancement[J]. IEEE Trans Acoust. Circuits Syst,1991,38:984-993.
- [6] Sun T, Neuvo Y. Detail-preserving median based filters in image processing[J]. Pattern Recognit Lett,1994,15:341-347.
- [7] Garcia-Cabrera L, Luque-Eacanmilla P L, Martinez-Aroza J, et al. Two pixel preselection methods for median type filtering[J]. IEEE Proc visual Image Signal Processing,1998,(1):30-40.
- [8] Werman M, Peleg S. Min-max operators in texture analysis[J]. IEEE Trans Pattern Anal Machine Intell,1985,(6):730-733.

(上接第 46 页)

[9] 刑藏菊,王守觉,邓浩江,等. 一种基于极值中值的新型滤波算法[J]. 中国图象图形学报,2001,(6):533-536.

[10] Piotr S. Fast Impulsive Noise Removal[J]. IEEE Transactions On Image Processing,2001,10(1):173-179.

(编辑:姚树峰)

## A New Impulsive Noise Removal Filtering Algorithm

ZHOU Rui - rui, BI Du - yan, MAO Bai - xin

(The Engineering Institute, Air Force Engineering University, Xi'an, Shaanxi 710038, China)

**Abstract:** A new filtering algorithm with the purpose of eliminating impulsive noise is presented. In order to keep the content of signals as much as possible when the noises are removed, the following approaches are developed in this paper. Noisy items are identified by inspection of their surrounding neighborhood, and afterwards their values are replaced with the most "conservative" ones out of their neighbors' value. If a pixel value is the extremum (max. or min.) of its neighborhood, it is a noise pixel, or else it is a signal pixel. Experiment results show that the new algorithm is good in performance to keep the content of image and remove noise, especially in the case of strong noise.

**Key words:** filtering; impulsive noise; noise elimination