ON THE MEDIAN FILTER METHOD

Bardachenko V. F., Koval'chuk L. V.

NAS Institute of Cybernetics
The Center of timer computing systems
40 Academic Hlushkov avenue,
Kyjiv
252022 Ukraine
tel. 266-51-68

ABSTRACT

This article is devoted to different questions, connected with median filter method (MFM). Several methods are described to increase the speed and to improve the accuracy in median filtration processes. Also algorithm is given to choose the optimal neighborhood for each pixel to process it by MFM.

1. INTRODUCTION

The median filter method (MFM) is one of the most popular methods used for removing "shot" noise from images and sound signals.

The median of the *2k-1* element set is the *k*-th element in the sorted list of *2k-1* elements. According to the MFM, the median value of all the pixel values from some array replaces the original center pixel. Denoting the function of the initial pixel values by *g(i,j)*, we obtain the new pixel values by the next formula:

$$m(i,j) = \text{median} \{ g(m,n), (m,n) \in W(i,j) \},$$

where *W(i,j)* is the neighborhood of the pixel with coordinates *(i,j)*. The shape of the neighborhood is chosen in depending of image shapes. The most popular neighborhoods are "window" or "cross".

MFM is often more preferable than average filter method (AFM), or linear filter method. MF is more convenient to remove impulse noise. It removes large deviations appearing with small probabilities much better than AF; in this cases variance estimates for MF are better. MF saves the relief borders of pictures when AF cleans the borders and makes them be loosed. MF uses only operation of comparison, hence, it does not give round-off errors that may appear during AF processing. Except that, MF is realized more simply because of using only logic elements, when AF uses the operations of addition and division.

Thanks to this advantages, MF is often used in seismic signal processing, speech processing, computer tomography, medical image processing, location of peaks and so on.

MF is realized more suitable using timer operands ([1]), because working with timer operands on the base of standard binary logic it is sufficiently easy to realize three important functions of logic processing of timer operands: finding of minimum, maximum and absolute value of difference.

2. ACCURACY IMPROVING

During the image processing by the MFM, we often have to solve the question: what neighborhood should be chosen for the processed pixel (see, for example, [2])? It is clear that this choice depends on the geometry properties of the image. But these properties may be different for different pixels.

So, we have to choose the neighborhood for each pixel individually. Except this, it is not always good to choose such neighborhood that the processed pixel is situated in its center. The results of image processing are often more exactly if it is situated somewhere else (for example, for "window 3x3" type neighborhood we can chose one of the 9 locations for the pixel). Especially it is important during the processing of the image borders or its long narrow parts. The following method gives us the rule how to chose the most suitable neighborhood for the given pixel.

Let us denote

p—the processed pixel;

i(p) – the number value that expresses the intensity (or the color) of the image in the pixel *p*;

V(p) – some neighborhood of it.

First of all we have to choose the set

$$U = \{ V_K(p), k = \overline{1,m} \}$$

of the neighborhoods to look for the most suitable one. For example, it may be the set that consists of all the "window 3x3" type neighborhoods (9 elements), all the "cross 3x3" neighborhoods (5 elements) and all the "cross 5x5" neighborhoods (9 elements). For each of this neighborhoods we denote the value that depends on the neighborhood, the pixel and the intensities (or colors) of the elements of the neighborhood.

Let the neighborhood *V(p)* has *n* elements $p_l, l = \overline{1,n}$ (including the pixel *p*). Then the value

$$d(p,V) = \frac{1}{n} \sum_{l=1}^{n} |i(p) - i(p_l)|$$

we call *the average deviation of the intensities* (or *the colors*) for the pixel *p* and the neighborhood *V*. Now we find $d(p,V_K(p))$ for each $V_K(p) \in U, k=\overline{1,m}$. And after that we choose the most suitable neighborhood $\overline{V}(p)$ such that

$$d(p,\overline{V}) = \min\{d(p,V_K),k=\overline{1,m}\}.$$

For the neighborhood $\overline{V}(p)$ we do processing of the pixel *p* by MFM.

This is the way to avoid the distortion of the image's borders and its long narrow parts.

3. SPEED INCREASING

Very often it is need to do image processing in real time. For this case we use truncated median (where several least significant digits are replaced by zeros) instead of full median. This expedient essentially decreases the number of quantization impulses in operand and doesn't give us a large error.

Let us assume that we clip least k significant digits in the calculating of *n*-digital numbers median. Then the maximum error $\delta(n,k)$ obtained in truncated median $M(n,k)$ is find by such expression:

$$\delta(n,k) = \frac{2^k - 1}{2^n - 1} \cdot 100\% \approx 2^{k-n} \cdot 100\%.$$

In the following table the values of $\delta(n,k)$ are given for $n=8$ and $n=16$.

K	N=8	n=16
1	0,39%	0,001%
2	1,17%	0,004%
3	2,75%	0,011%
4	5,88%	0,023%
5	12,16%	0,047%
6	24,71%	0,096%
7	49,80%	0,19%
8		0,39%
9		0,78%
10	———	1,56%
11	———	3,12%
12	———	6,25%
13	———	12,50%
14		25,00%
15		50,00%

When we use truncated median $M(n,k)$ instead of full median $M(n) = M(n, 0)$, we essentially decrease the number of quantization impulses.

Denote $i(n)$ the number of quantization impulses for full median $M(n)$ and $i(n,k)$ the number of quantization impulses for $M(n,k)$. Then we obtain such formula:

$$i(n,k) = \frac{2^k - 1}{2^n - 1} i(n)$$

For example, the number of quantization impulses for $M(8, 4)$ is smaller in 17 times; the number of quantization impulses for $M(8, 6)$ is smaller in 85 times in compare with $M(n)$.

Substituting $n=k+1$ in the last formula, we obtain that the maximum error is about 50% of the last significant digit.

The value of truncated median is always smaller than the value of full median. Using this fact we can decrease the value of the maximum error. If we take

$$M(n,k) + 2^{K-1}$$

instead of truncated median $M(n,k)$, the value of maximum error will decrease in two times. It should be noted that during black-and-white image processing by truncated median it is possible to exclude error completely.

4. CONCLUTION

In conclusion we want to say about two others methods to increase the speed in MF-processes: to use "speed" sorting nets with non-parallel connections (see [3]) and to devise the operands into several groups of bits (see [4]).

REFERANSES

1. Bardachenko V.F. "Conceptual foundation of the timer measuring and computing devices and systems" Methodical directions for studying of the course "Measuring and computing devices and systems". Kiev KPI 1995, 63p.
2. Rytsar Y. B., Ivasenko I. V., Alexiychuk O. V." A structure-algorithm of the filtration of additive noise from images." Lviv, The National Academy of Science of Ukraine, physics- mechanical Institute named by Karpenko-Karyi, "Vidbir i obrobka informatcii",1997, no 11, p.103-107.
3. Bardachenko V.F., Koval'chuk L.V." "On the truncated median method." Kiev, KPI,"Measurement-98", p.150-152.
4. Koval'chuk L.V." Some methods of speed increasing in median filtration processes". Kiev, USIM, 1998, no. 2, p.13-16.