Histogram-based weighted median filtering used for noise reduction of digital elevation model data

Roland Kilik¹

Received: 25 May 2021 / Accepted: 1 August 2021 / Published online: 26 August 2021
© The Author(s) 2021

Abstract

A new histogram-based robust filter developed for noise reduction of digital elevation model data is presented. When large percentage of data points in data matrices are contaminated with outlier noise, the noise reduction process can give better results than traditional median filtering, if elements with a potentially higher chance of being noise are eliminated by weighting from the input dataset before the median value is calculated. However, on the same matrices, there are likely to be subsets of data where unfiltered input is more reasonable for the calculation. The new method implementing weighting between these two cases is presented below, with its initial tuning and a comparison with both standard median filtering and the Most Frequent Value (MFV) method, as the latter being much more efficient than the usual methods. Following the description of the procedures, their effectiveness is compared for noise reduction in digital elevation model data systems, at various noise levels. The comparison is done mainly by three measures, with most of the focus on the L_1 norm data distance results. Finally, a modified version of the method—which includes Steiner’s MFV filter as a core part—is also introduced, with similar examination. The method to be presented has been shown to be superior to conventional median filtering for most noise rates, and in many cases also to Steiner’s MFV, for handling non-zero mean noises. The modified version of the method—with the help of Steiner’s MFV—has also achieved this in handling zero mean noise, in the field of application described in the paper.

Keywords Noise reduction · Median filter · Digital elevation model · Weighted median

1 Introduction

Due to its robustness, the Most Frequent Value (MFV) method (Steiner 1991) can be well applied in the processing of noisy datasets in geophysical (Dobróka et al. 1991; Szabó and Balogh 2016; Szabó and Balogh 2018), hydrogeological research (Szűcs and Zákányi 2007) and various other fields (Zhang 2017). Such data systems, especially those with extreme noise, are also present in the field of spatial informatics, including digital elevation

✉ Roland Kilik
gfrk@uni-miskolc.hu

¹ Department of Geophysics, University of Miskolc, Miskolc-Egyetemváros 3515, Hungary

modelling and satellite transmission. A similarly widely used, but less robust and sophisticated technique is the median filtering method (Stone 1995; Huang et al. 1979).

In this paper, a new median filtering method, improved by histogram operations and weighted averaging is presented, and compared to the original median filter and the MFV based procedure. The method's modified and also presented version – which aims to eliminate zero mean noises – contains Steiner's MFV filter as a core part.

The aim of the proposed method's both version is to eliminate scattered noise from digital elevation models in a moving windowed manner (i.e. the procedure corrects the central element of the actual window). This study was also conducted for noise exposure at four different percentages of data points.

2 Input dataset

The analysed data consisted of three 25 m spatial resolution digital elevation models of different areas, created using Topo to Raster interpolation in ArcGIS software by digitizing the contour lines, elevation points and water network of 1:10,000 scale EOTR map sheets. The histograms of the three datasets can be seen in Fig. 1. The mean of the data in the first dataset is 211.99, while the standard deviation is 8.440. The same statistical data for the second dataset is 191.31 mean and 8.752 standard deviation, while value of the mean is 96.39 and the standard deviation is 0.457 in the third data system.

For each of the resulting digital elevation model data systems, firstly normally distributed noise added to the data matrices, with a standard deviation to have the average noise amplitude at around 1% of the mean of the data matrix. After this—as outlier non-zero mean impulse noise—additional noise was added to 10, 15, 20, and 25 percent of the points randomly. In order to achieve this, a normally distributed noise vector is generated for every row with the mean equal of the mean of the given data row, and standard deviation to have the average noise amplitude around 100% of the mean of the data row.

Then, the elements of the noise vector generated for a given row of data were randomly scattered across the row with a multiplier between 0.1 and 0.7, giving an additional ~10–70% noise to the data in the different test cases (referred as 0.1–0.7 noise amplitude below).

3 Introduction of the weighted median (WM) method

The method is used to produce the corrected value of the central element at each position of the moving window going through the image matrices using a weighted mean. The following version of the presented method is mainly for eliminating non-zero mean noises (due to, for example, measurement device problem or long distance data transfer).

The weighted mean is calculated for each data point, with two weights (w_1 , w_2) defined below. In order to achieve this, two independent window narrowing process occurs as an initial step, before calculating the weights. These narrowed windows are created from the actual (5×5) data window, at every window position. As a test, a window size of 9 (3×3) and 49 (7×7) was also applied, but this did not prove to be optimal for solving the problem.

The process of the first window narrowing is the following.

The range of the values of the elements in the moving window is divided into two and three bins with equal range widths, and then two ratios are generated:

Fig. 1 Histograms of input data matrices

- λ_1 the element count ratio between the larger and the smaller element count domain out of the two,
- λ_2 ratio of the largest and second largest domains out of 3 (regarding the element count again).

If $\lambda_1 > \lambda_2$, the new set (D) is defined as the most element count domain of the 2 domains, otherwise as the most element count domain of the 3 domains. Then the value of m_s defined as

$$m_s = \text{median}(D).$$

Thus, we used higher λ value as an indication of a sharper cut. The ideal number of 2 and 3 as the number for ranges chosen because in case of splitting into 4 (or more), a bin would possibly not contain a sufficient number of elements from the initial 5×5 window, in order to do the further steps described.

Then another (independent) second window narrowing process occurs, for determining the value of parameter m_{e1} . In order to achieve this, the original moving window's elements are sorted by value, then divided into two and three equal width ranges (based on the set of values).

For example, in the case of splitting into three, if the ordered vector is $\mathbf{v}$, and $\max(\mathbf{v})$ is its highest value element, and $\min(\mathbf{v})$ is the lowest, then in the third with the lowest values will be the values lower than $\min(\mathbf{v}) + (\max(\mathbf{v}) - \min(\mathbf{v}))/3$.

Here we calculate ratio λ_3 from the case when splitting into two. The value of λ_3 is $1/n$, where n is the sum of element count of the two bins without the highest valued bin (i.e. the bin with the lower values). Then we calculate λ_4 ratio from the case when splitting into three bins. The value of this ratio is $1/m$, where m is the sum of the element count of the three bins without the highest valued bin.

If $\lambda_3 > \lambda_4$: we take the highest valued bin from the two (the bin with the highest values), otherwise we take the highest valued bin from the three as the chosen set (E). Finally, m_{e1} will be the average of the elements of the chosen new set.

A similar value, m_{e2} is determined with the same method, however by splitting the original window into 3 and 5 parts (instead of 2 and 3). With this step, a higher division number is reflected in the result, if the current window's value set allows it (i.e. if the new intervals' element count is not zero).

As m_{e1} and m_{e2} are both calculated on chosen subsets of the largest values from the value set of the window, both m_{e1} and m_{e2} are related to maximums. Important difference between m_e and m_s that in m_{e1}, m_{e2} the mean is calculated while in m_s the median is calculated, with different bin numbers. In addition, in m_{e1} and m_{e2} , the narrowest set has the largest valued elements, and in m_s , the set contains the largest number of elements (which may not necessarily contain the largest values).

Now we can calculate the first weight (w_1) of the current point's weight vector as

$$w_1 = (m_s/m_{e1}) * \alpha, \quad (1)$$

where α is a scaling factor that ensures that the value obtained by w_1 falls within the same range of values as w_2 , described below. Its value ($\alpha = 1/3$) is determined on experimental basis to fulfill this purpose.

We can also define m_w as the median of the values in the original moving window.

Using the values described above, three sub-weights are produced as follows (all will have a role in determining the value of w_2 weight).

$$w_a = \frac{1}{\max_1} * (m_{e1} - \max_1) + 1, \quad (2)$$

where $\max_1: \max(m_w, m_{e1})$,

$$m_{as} = |m_{e1} - m_{e2}|, \quad (3)$$

$$w_p = \frac{\beta}{\max_2} * (m_{as} - \max_2) + 1, \quad (4)$$

where $\max_2: \max(m_w, m_{e1}, m_{e2})$,

$$w_{p2} = \frac{\gamma}{\max_3} * (m_{as} - \max_3) + 1, \quad (5)$$

where $\max_3: \max(m_w, \text{mean}(\text{window}))$.

The calculation of the max value applied in a given sub-weight (w_a, w_p, w_{p2}), shall in all cases include the median of the window, as well as the mean (m_{e1}, m_{e2}) or median (m_s) value of the narrowed window.

In Eqs. 4 and 5, fewer elements are omitted from the original window (because of containing m_{e2} via m_{as}), so that the sub-weights w_p , and w_{p2} both calculated with a smaller multiplier than in Eq. 2. The values of these β and γ constants are chosen as 0.5 (the adjustment procedure's results can be in seen Sect. 6).

Both w_p and w_{p2} weights have a corrective role. Their value can be high if there is a large difference between the averages of the subsets obtained by splitting the weights into 3 and 5 (m_{e1} and m_{e2}). As can be seen in Fig. 2, a large difference between the values of m_{e1} and m_{e2} results in a large L_1 norm error. Thus, a large difference indicates that the histogram operations in the current window position may distort the result, so increasing value

Fig. 2 L_1 norm and m_{as} value relation

of the difference increases the weight of w_2 (i.e. the conventional median without histogram operations).

The subweight w_a can take a high value other than 1, if the median value of the original, unconstrained window m_w is greater than the average of the elements of the constrained window (m_{e1}). Because the narrowed window contains the largest values of the subsets, if the difference between the median of the original window and the mean of the elements of this narrowed window is outstanding, it indicates that histogram operations at the current window position distorting the result. As in the previous, a large positive difference between m_{e1} and max_1 results in a large L_1 norm error.

Therefore, this should be reflected in the final weight vector $\mathbf{w}$ in the form of either a reduction in the value of w_1 or an increase in the value of w_2 (i.e. an increase in the weight of the result of the traditional median method). The latter is achieved with the usage of w_a in weight w_2 . Since w_a is the most important of the three correction factors (w_a , w_b , w_c), its square is included in the formula w_2 . The higher effect was achieved by squaring the weight, because the maximum value of the weight before adding one to its value is 1, so squaring the weight will not result in an extreme weight value, even at its maximum.

The + 1 in the formulas for the partial weights w_p and w_{p2} is included because they all include a maximum value subtraction, which in most cases results in a negative value, so the constant provides a shift into the positive range. In the formula of w_a , the role of adding 1 is shifting its minimum to greater than one (in order to be able to increase w_2 weight in its squared value).

Finally, the following two weights (w_b , and w_c) are produced using w_p and w_{p2} respectively.

$$w_b = 1 + \frac{w_p}{2}, \quad (6)$$

where m_w is the median of the values in the original moving window. Since the maximum value of the partial weight w_c is not a function of the different narrowed windows, but of the median or average value of the original window, this partial weight is taken with a smaller constant:

$$w_c = 0.5 - \frac{w_{p2}}{2}. \quad (7)$$

With the components defined above, the weight w_2 takes the following form

$$w_2 = w_a^2 * w_b * w_c. \quad (8)$$

At this point, we know the $\mathbf{w}$ weight vector of the current data point

$$\mathbf{w} = [w_1 w_2]. \quad (9)$$

In weight vector $\mathbf{w}$ (Eq. 9), weights w_1 and w_2 have an effect on the median of the current data window (m_w) on the one hand, and on the median of the reduced set of the same window (m_s) on the other hand, w_1 weighting the latter, and w_2 weighting the former as follows (for example, on the k -th element of the data matrix):

$$res_{WMk} = (w_1 * m_s + w_2 * m_w) / (w_1 + w_2). \quad (10)$$

As described above, the median of a narrowed window (m_s), and the original window's median (m_w) is weighted at every window position for the final result of the

actual point. Weight of m_s is $(m_s/m_{e1}) * \alpha$ (i.e., the median is divided by the average of the maximal values of narrowed window). If this ratio is for example low for the given moving window position due to the noise (high m_{e1} average of maximums), then m_s 's weight should be proportionally low, because otherwise, the high value of the outlier maximums would have negative effect on the final result. In such cases m_w 's weight will be proportionally high – not only because of the low weight of m_s , but due to the fact that m_w is weighted by w_a, w_b, w_c , all containing m_{e1} or m_{e2} values.

4 Most Frequent Value Method

A much more reliable statistical characteristic than the arithmetic mean, the weighted mean, is obtained by assigning a small weight (w_k) to points far from the majority of the data (X_k) and a larger weight (w_k) to points in the highest data density location (Eq. 11).

$$M = \sum_{k=1}^N X_k w_k \left[\sum_{k=1}^N w_k \right]^{-1} \quad (k = 1, 2, \dots, N). \quad (11)$$

The k -th weight is chosen by Steiner (Steiner 1991) as follows:

$$w_k = \varepsilon^2 / \left[\varepsilon^2 + (X_k - M)^2 \right]. \quad (12)$$

In the above, N is the number of data and ε is the dihesion, a scalar parameter. If ε is large, then all data are given nearly equal weights and outliers will spoil the value estimate, and if ε is too small, care must be taken to avoid ignoring some data.

The weighted mean, called the most frequent value (M) defined by Eq. (11), should be known in advance in order to assign weights with maximum values at its location and smaller and smaller weights away from it. Therefore, this procedure requires an iterative algorithm in which M and ε are determined jointly. In the first iteration step, the dihesion can be estimated from the sample space using the following formula:

$$\varepsilon_1 = \left(\sqrt{\frac{3}{2}} \right) [\max(X_k) - \min(X_k)], \quad (13)$$

while for the M_1 the initial value is preferably chosen as the sample mean or median. In this study, the median value was used.

In subsequent iteration steps, M and ε can be derived from each other according to the following procedure:

Table 1 Example of adjusting constant value used for calculating w_p

β	2	1.5	1	0.5	0.25	0.125	0.0625	0.03125
L_1 norm distance	843.12	822.32	801.63	762.82	794.51	956.47	1720.47	2393.27

$$\varepsilon_{j+1}^2 = \frac{3 \sum_{k=1}^N \frac{(X_k - M_j)^2}{[\varepsilon_j^2 + (X_k - M_j)^2]^2}}{\sum_{k=1}^N \frac{1}{[\varepsilon_j^2 + (X_k - M_j)^2]^2}} \leftrightarrow M_{j+1} = \frac{\sum_{k=1}^N \frac{\varepsilon_{j+1}^2}{\varepsilon_{j+1}^2 + (X_k - M_j)^2} X_k}{\sum_{k=1}^N \frac{\varepsilon_{j+1}^2}{\varepsilon_{j+1}^2 + (X_k - M_j)^2}}. \quad (14)$$

5 Description of the resulting numerical values

For comparing the results of the different filtering methods, the following metrics are used in the paper. Calculation of the RMSE (Root Mean Square Error) value for the MFV method and median filtering (where inp is the noise-free data matrix):

$$RMSE_{St} = \sqrt{\frac{\sum_{i=1}^N (res_{Sti} - inp_i)^2}{N}}, \quad (15)$$

where res_{St} : matrix corrected by Steiner's MFV,

$$RMSE_{Med} = \sqrt{\frac{\sum_{i=1}^N (res_{Medi} - inp_i)^2}{N}}, \quad (16)$$

where res_{Med} : matrix corrected by median method,

$$RMSE_{WM} = \sqrt{\frac{\sum_{i=1}^N (res_{WMi} - inp_i)^2}{N}}, \quad (17)$$

where res_{WM} : matrix corrected by weighted median.

Deviation regarding the three procedures:

$$Std_{St} = \overline{std(res_{St} - inp)}, \quad (18)$$

$$Std_{Med} = \overline{std(res_{Med} - inp)}, \quad (19)$$

$$Std_{WM} = \overline{std(res_{WM} - inp)}, \quad (20)$$

where res_{St} , res_{Med} , res_{WM} , inp : as above.

L_1 norm:

$$L_{1St} = \| res_{St} - inp \|_1, \quad (21)$$

$$L_{1Med} = \| res_{Med} - inp \|_1, \quad (22)$$

$$L_{1WM} = \| res_{WM} - inp \|_1, \quad (23)$$

where res_{St} , res_{Med} , res_{WM} , inp : as above.

6 Adjusting constants

Regarding the 0.5 constant value of w_p , it was also tested in some randomly chosen test cases, that how the L_1 norm distance value between the noise-free matrix, and the weighted median-corrected matrix changes with the different values of the constant. The value of the norm monotonously decreased in every such cases, as can be seen as an example in Table 1 (10% noise rate, 0.3 noise amplitude):

Similarly, the 0.5 constant of w_{p2} was also tested, and it produced similar results, as can be seen in the example of Table 2 (for the same parameters as in the previous case):

All of the constants are examined with a few chosen values, however, the global optimisation of them is not part of this paper.

7 Comparative results

In all cases, a 5×5 window ran through the matrices, and both the Steiner-MFV method (using as a filter – Dobróka 2021) and the weighted median method described here always corrected the value of the window's central element, with all elements of the window as input.

In all cases, the number of iterations for the Steiner filter was 20 and the initial weight was the median of the window elements.

In all cases the results were also compared with those obtained using the classical median filter in MATLAB software. The median filter ran on the same noisy data matrices with the same window size as the Steiner method and the weighted median method.

The example shown in Table 3 shows the values of the result metrics on the first data series, with noise on 15% of the data points, with a noise amplitude multiplier of 0.3

Another example can be seen in Fig. 3, again noise on 15% of data points, now with a noise amplitude multiplier of 0.5. Figure 3. (a) shows the original data, (b) the noisy data, (c) is the result of Steiner method, (d) is the result of the weighted median method and (e) is the result of the classical median method.

Table 4 shows the values of the L_1 norms showing the distance from the noise-free input data matrix and their ratios, using the weighted median method (L_{1WM}) and the Steiner method (L_{1St}), with 25% of the points contaminated with noise, as a function of different noise amplitudes (0.1,...,0.7) on the first data set. The values show that in two cases the Steiner method gives better results, by about 6%, and in the other cases the weighted median method proved to be better. The latter gives better results by 6.3% on average (since the average of the L_{1WM}/L_{1St} ratios is 0.937).

Table 2 Example of adjusting constant used for calculating w_{p2}

γ	2	1.5	1	0.5	0.25	0.125	0.0625	0.03125
L_1 norm distance	735.13	721.78	698.67	654.31	668.807	668.94	676.11	685.71

Table 3 Example of comparative results (10% additional outlier noise on 15% of the data points)

L_{1WM}	538.98
L_{1St}	630.41
L_{1Med}	711.19
$RMSE_{WM}$	1.42
$RMSE_{St}$	2.02
$RMSE_{Med}$	2.13
Std_{WM}	1.19
Std_{St}	1.75
Std_{Med}	1.58

Table 5 shows the results of the weighted median procedure and the standard median filtering at the same noise level. In this case, the weighted median procedure proved to be better on the data set by 26.4% on average.

Tables 6 and 7 show the same comparison as before, regarding the L_1 norm for noise affecting 20% of the points. In this case, there are noise amplitude values where the Steiner method gives a smaller distance to the noise-free matrix than the weighted median procedure, but in none of the cases the standard median procedure could achieve this. The weighted method is on average 23% better than the latter.

Tables 8 and 9 show the case with 15% noisy points.

The results of Table 8 comparing the Steiner method with the weighted median procedure show that for a noise amplitude multiplier of 0.1, the Steiner method is superior.

The weighted median procedure outperforms the unweighted median procedure by an average of 21.2% on the first data set for 15% noisy points (Table 9). Tables 10 and 11 show the data distances of the three procedures for the case where 10% of data is contaminated by noise according to the L_1 norm.

In this case (with noise on 10% of the points), the Steiner method outperforms the weighted median method in most of the different noise amplitudes—in 4 cases, with an average of 5.6% in the four cases, and in the remaining three cases the WM method outperforms on the data set, by 13%.

Table 12 shows the RMSE values obtained by the three procedures and their ratios, for a noise exposure of 25% on the left and 20% on the right, for different noise amplitude multipliers (0.1, ..., 0.7) in both cases.

Table 13 shows the RMSE values as before, now for noise at 15% and 10% of the data points.

It can be seen that the weighted median method performs worse for the highest noise amplitude multipliers (0.7), but this is also true for the other methods, so the ratio to the other methods does not deteriorate. The RMSE obtained by the MFV method is closest to that obtained by the weighted median method for the smallest noise amplitude multiplier (0.1). The RMSE value for the weighted median method is on average 79.5% of that obtained by the conventional median method.

As the weighted median procedure was found to be the least efficient when 10% of the data points were contaminated with noise in the above tests, the standard deviations were also examined in this case. An example of this compared to the standard median method is shown in Table 14.

Table 15 shows the average of the results for the first and second data sets. The minimum, maximum and average of the data distance ratios for the L_1 norm as a function of the

Fig3 Visual example of results on the first dataset

Table 4 L1 norm values at 25% noise ratio

Noise amp. mult	0.1	0.2	0.3	0.4	0.5	0.6	0.7
L_{1WM}	540.22	425.44	420.39	471.26	615.92	787.36	515.82
L_{1St}	597.90	457.46	536.20	541.64	580.61	738.81	545.17
L_{1WM}/L_{1St}	0.90	0.92	0.784	0.87	1.06	1.06	0.94

Table 5 L1 norm values at 25% noise ratio

Noise amp. mult	0.1	0.2	0.3	0.4	0.5	0.6	0.7
L_{1WM}	540.22	425.44	420.39	471.26	615.92	787.36	515.82
L_{1Med}	604.39	582.33	725.92	692.10	827.83	970.98	723.63
L_{1WM}/L_{1Med}	0.89	0.73	0.57	0.68	0.74	0.81	0.71

Table 6 L1 norm values at 20% noise ratio

Noise amp. mult	0.1	0.2	0.3	0.4	0.5	0.6	0.7
L_{1WM}	566.65	437.97	440.14	485.61	708.77	646.18	759.37
L_{1St}	545.34	468.96	469.76	608.38	1002.99	866.07	841.60
L_{1WM}/L_{1St}	1.03	0.93	0.93	0.79	0.70	0.74	0.90

Table 7 L1 norm values at 20% noise ratio

Noise amp. mult	0.1	0.2	0.3	0.4	0.5	0.6	0.7
L_{1WM}	566.65	437.97	440.14	485.61	708.77	646.18	759.37
L_{1Med}	612.87	627.87	612.76	741.06	901.39	792.62	979.72
L_{1WM}/L_{1Med}	0.92	0.69	0.71	0.65	0.78	0.81	0.77

Table 8 L1 norm values at 15% noise ratio

Noise amp. mult	0.1	0.2	0.3	0.4	0.5	0.6	0.7
L_{1WM}	542.97	433.71	538.98	464.25	493.81	675.11	624.08
L_{1St}	519.85	473.9043	630.41	490.20	561.66	732.78	644.35
L_{1WM}/L_{1St}	1.04	0.91	0.85	0.94	0.87	0.92	0.96

different noise levels can be seen for the Steiner method and the weighted median method. The mean of the minima (i.e. the mean of the cases where the largest difference is in favour of the weighted median method) is 0.82, i.e. in these cases the method is 18% better. The average of the maxima is 1.032, i.e. in the opposite case the Steiner method is on average 3.2% better for the two data systems combined.

Table 9 L_1 norm values at 15% noise ratio

Noise amp. mult	0.1	0.2	0.3	0.4	0.5	0.6	0.7
L_{1WM}	542.97	433.71	538.99	464.25	493.81	675.11	624.08
L_{1Med}	604.28	614.34	711.192	639.07	672.32	846.64	695.19
L_{1WM}/L_{1Med}	0.89	0.7	0.75	0.72	0.73	0.79	0.89

Table 10 L_1 norm values at 10% noise ratio

Noise amp. mult	0.1	0.2	0.3	0.4	0.5	0.6	0.7
L_{1WM}	521.53	483.07	543.86	484.36	779.51	727.48	724.2
L_{1St}	507.13	482.52	542.96	527.25	652.87	939.2	790.58
L_{1WM}/L_{1St}	1.02	1	1	0.91	1.19	0.77	0.91

Table 11 L_1 norm values at 10% noise ratio

Noise amp. mult	0.1	0.2	0.3	0.4	0.5	0.6	0.7
L_{1WM}	521.53	483.07	543.86	484.36	779.51	727.48	724.2
L_{1Med}	605.38	637.25	746.44	645.17	871.46	1038.26	865.18
L_{1WM}/L_{1Med}	0.86	0.75	0.72	0.75	0.89	0.7	0.83

Table 16 shows the L_1 norm results grouped according to the same method, in this case for the two median procedures. The average of the minima is 0.692, so in the cases where the weighted median procedure is the best, this method is better by more than 30%. The average of the maxima is 0.86, so that even in the worst cases the weighted median procedure is on average 14% better than the standard median filtering for the two data sets combined.

A less detailed comparison was carried out on the third set of data (examining only the L_1 norm ratios). This study showed similar characteristics as the previous ones, however the Steiner method was found to be the best of the three in more cases than before. For 10% and 15% noisy data points (both with 7 different noise amplitudes, as before), the Steiner method was found to be superior to the weighted median method in 11 out of 14 cases, with an average of 14.6% (regarding L_1 norm ratios). In the 20% and 25% noisy point cases, the weighted median method gave a better result, in 14 out of 14 cases, with an average of 19.04%.

Comparing the two median methods on the data set (again with L_1 norm ratios), in 27 out of 28 cases, the weighted median method proved to be better, with an average of 14.32%.

8 Handling zero mean noises

The previously presented version of the proposed method developed for mainly non-zero mean noises. In the following, a second, modified version of the method is introduced, what purpose is mainly handling zero-mean normal distribution noises. This version of the

Table 12 RMSE values at 25% and 20% noise ratio

Noise ratio	25%							20%						
Noise amp. mult	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.1	0.2	0.3	0.4	0.5	0.6	0.7
RMSE _{NM}	1.54	1.47	1.55	2.04	2.50	2.73	2.25	1.58	1.51	1.71	1.63	2.14	2.52	2.74
RMSE _{NMed}	1.79	1.95	2.16	2.50	2.84	3.07	2.64	1.84	2.03	2.30	2.17	2.58	2.95	3.10
RMSE _{St}	1.60	1.74	2.07	2.73	3.29	3.79	3.76	1.63	1.82	2.21	2.50	3.09	3.61	4.24
RMSE _{NM} /RMSE _{NMed}	0.86	0.75	0.72	0.82	0.88	0.89	0.85	0.86	0.75	0.74	0.75	0.83	0.85	0.88
RMSE _{NM} /RMSE _{St}	0.96	0.84	0.75	0.75	0.76	0.72	0.60	0.97	0.83	0.77	0.65	0.69	0.70	0.65

Table 13 RMSE values at 15% and 10% noise ratio

Noise ratio	15%							10%						
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.1	0.2	0.3	0.4	0.5	0.6	0.7
Noise amp. mult														
RMSE _{WM}	1.59	1.55	1.42	1.70	1.83	2.34	2.55	1.60	1.46	1.75	1.50	2.12	2.02	2.72
RMSE _{Med}	1.83	2.04	2.13	2.22	2.36	2.74	2.92	1.86	2.00	2.29	2.10	2.56	2.60	3.08
RMSE _{St}	1.62	1.83	2.02	2.55	2.97	3.72	3.81	1.66	1.84	2.13	2.36	3.01	3.32	4.23
RMSE _{WM} /RMSE _{med}	0.87	0.76	0.67	0.77	0.77	0.85	0.87	0.86	0.73	0.77	0.72	0.83	0.78	0.88
RMSE _{WM} /RMSE _{St}	0.98	0.85	0.70	0.67	0.61	0.63	0.67	0.96	0.80	0.82	0.64	0.71	0.61	0.64

Table 14 Standard deviation values at 10% noise regarding the two median methods

Noise amp. mult	0.1	0.2	0.3	0.4	0.5	0.6	0.7
Std_{WM}	1.41	1.25	1.37	1.20	1.43	1.37	1.62
Std_{Med}	1.42	1.51	1.70	1.56	1.76	1.74	1.96
$\text{Std}_{\text{WM}} / \text{Std}_{\text{Med}}$	0.99	0.82	0.80	0.77	0.81	0.79	0.83

Table 15 L1 norm ratios for two data sets combined

Noise exposure percent-age	$\min(L_{\text{IWM}}/L_{\text{ISt}})$	$\max(L_{\text{IWM}}/L_{\text{ISt}})$	$\text{avg}(L_{\text{IWM}}/L_{\text{ISt}})$
25%	0.81	1.07	0.93
20%	0.81	0.99	0.91
15%	0.83	0.98	0.90
10%	0.80	1.07	0.93

Table 16 L1 norm ratios for two data sets combined

Noise exposure percent-age	$\min(L_{\text{IWM}}/L_{\text{IMed}})$	$\max(L_{\text{IWM}}/L_{\text{IMed}})$	$\text{avg}(L_{\text{IWM}}/L_{\text{IMed}})$
25%	0.63	0.86	0.74
20%	0.67	0.86	0.77
15%	0.73	0.84	0.78
10%	0.72	0.85	0.77

method calls upon and uses Steiner's MFV values to correct the actual central element of the given data window.

8.1 Noise generation

Regarding the noise generation process, in the first step, a general zero mean noise was added to the data matrix. In order to achieve this, a normal distribution noise generated for every row of data, with the mean value equal to the mean of zero, and standard deviation 1.

In order to add outlier noise to given percentage of points for the examination, additional zero mean normal distribution noise is added randomly to 20, 15, 10, and 5 percent of the points, with 0.1–0.7 amplitude multiplier in all of such cases (as in the previously introduced version of the method). The noise's standard deviation was always the mean of the current data row.

8.2 Modified version of the weighted median method

As with the method's previously introduced version, the first steps are histogram filterings.

Firstly a median, then and a mean value is generated from filtered windows, however here with a two-step filtering for both. For producing the value of m_s , median value, the histogram-based filtering is the following.

Based on the set of values, the elements of the current data window are divided into two and three ranges (bins) with equal range widths, and then two ratios are generated:

- λ_1 : the ratio of the largest and second largest domains out of 2,
- λ_2 : ratio of the largest and second largest domains out of 3 (in both cases regarding the element count).

If $\lambda_1 > \lambda_2$, the new set (D) will have the most element count domain of the 2 domains, otherwise it will have the most element count domain of the 3 domains. Finally, the value of m_s is: $\text{median}(D)$.

For determining the value of m_{e3} , a second narrowing process is done. In this process, the window elements are first sorted by value and then divided into three equal width ranges based on the set of values. Here we calculate λ_3 ratio, as the ratio between the element count of the window and the sum of the element count of the thirds without the highest valued third. Then we calculate λ_4 , whose value will be the ratio between the element count of the most and the second most element count bins. Thus, λ_3 and λ_4 are calculated differently than in the previous version of the method.

If $\lambda_3 > \lambda_4$: we take the highest third (the bin with the highest values), otherwise take the bin with the most element count as the chosen set (E). Thus, we used higher λ value as an indication of a sharper cut.

We take this truncated E set, and distribute its values into bins. The bin width is determined with Scott's rule (Scott 1979, 1992):

$$3.5 * \text{std}(E) / \text{numel}(E)^{1/3}. \quad (24)$$

We must also determine the number of bins, in order to be able to distribute all the values into them (what is a trivial step because of having the bin widths and the data values). We take the bin with the largest element count, and m_{e3} is the average of the elements of the chosen bin.

Since we have the values of m_s and m_{e3} , we can replace the actual window's middle element with m_s (forming w_{r_ms}), and similarly with m_{e3} (constructing w_{r_me}). Finally, let the window with the MFV method's result at its centre w_{r_St} .

In the next step, we concatenate w_{r_ms} , w_{r_me} and w_{r_St} one by one with the original (noisy) actual window, forming w_{u1} , w_{u2} , w_{u3} .

Now we can calculate three gradient measures in the following way:

$$G(x, y) = \sqrt{\frac{\partial w_u}{\partial x} * \frac{\partial w_u}{\partial x} + \frac{\partial w_u}{\partial y} * \frac{\partial w_u}{\partial y}}, \quad (25)$$

$$g = \frac{1}{|X| * |Y|} \sum_{x \in X} \sum_{y \in Y} G(x, y). \quad (26)$$

In the formula, g_1 is the result of g when using w_{repl_St} , g_2 is the value of g in case of w_{repl_ms} , and g_3 is the value of g when using w_{repl_me} .

If $\min(g_1, g_2, g_3)$ is g_1 , then in the actual $\mathbf{w} = [w_1 w_2]$ weight vector, the value of w_1 is 0, and the value of w_2 is 1. Thus, in this case only the result of the MFV method counts in the given data window's correction. If $\min(g_1, g_2, g_3)$ is g_2 , then w_1 is 0.15 and w_2 is

Table 17 L1 norm ratios on first data set

	25% Noise exposure				20% Noise exposure			
	L_{1WM}/L_{1St}	L_{1WM}/L_{1Med}	L_{1WM}/L_{1St}	L_{1WM}/L_{1Med}	L_{1WM}/L_{1St}	L_{1WM}/L_{1Med}	L_{1WM}/L_{1St}	L_{1WM}/L_{1Med}
15% Noise exposure	0.987	0.982	0.993	0.978	0.991	0.988	0.978	0.979
10% Noise exposure	0.987	0.982	0.993	0.978	0.991	0.988	0.978	0.979
5% Noise exposure	0.987	0.982	0.993	0.978	0.991	0.988	0.978	0.979
2.5% Noise exposure	0.987	0.982	0.993	0.978	0.991	0.988	0.978	0.979
0% Noise exposure	0.987	0.982	0.993	0.978	0.991	0.988	0.978	0.979

[illegible]

Table 19 L1 norm ratios on third data set

[illegible]

0.85. If $\min(g_1, g_2, g_3)$ is g_3 , then w_1 is 0.4 and w_2 is 0.6. Thus, in this case, the weight of the MFV method's result is 0.6 for the given window.

As we can see, in all of these cases, we are weighting the Steiner MFV method's results, and increasing or decreasing its weight in the correction of the actual data window's central element.

Similar to the previous version of the method, we get poor results, if the value of m_{as} i.e. $|m_{e1} - m_{e2}|$ is large. In order to be able to handle this, here we have to calculate both m_{e1} and m_{e2} (the same way as in the previous version), and if the difference is greater than 2% of the mean of raw data, then w_1 should be 0, and the value of w_2 should be 1.

8.3 Results of the modified version of WM filtering procedure

In Table 17, L_1 norm ratios can be seen for the first data set, comparing WM method's results with both the MFV's result and the original median method's. In the former case, the WM method performed better in 26 test cases out of the 28. In that cases, the average of the L_1 norm ratio was 0.985, thus usage of the method resulted in 1.4% lower L_1 norm values on average. In the remaining two cases, the mean of the ratio was 1.005, thus the WM method performed 0.5% worse in those two cases. The best L_1 norm ratio value was 0.97, thus the WM method gave 3% lower L_1 norm value in that particular noise reduction.

Regarding the comparison with the original median method, the WM method performed better in all of the 28 cases, by 16.4% on average (0.836 average L_1 norm ratio). Here the best result was a 29.5% improvement (0.705 L_1 norm ratio).

Table 18 shows results in the same structure as the previous one, here on the second data set. Comparing with Steiner's MFV, the WM method gave better results according to L_1 norm in 23 cases (by 1.25% on average), and in the remaining 5 cases, performed worse by 0.21% avg. At its best, the WM method gave lower L_1 norm value by 3.12%.

In comparison with the other median method, WM performed better in all of the cases (16% avg., 29.4% max.).

In Table 19 the third data set's L_1 norm ratios can be seen. Comparing WM's results with the MFV method's, the former performed better in 23 of the 28 cases (by 1% avg., 2.4% max.), and the MFV method was superior in 5 cases (1.4% avg.).

Regarding the comparison with conventional median method, WM performed better in all of the cases, by 10.7% avg., 20.2% max.

9 Conclusions

The effectiveness of the histogram-based weighted median procedure described above has been demonstrated for noise elimination in digital elevation model data. The method's main purpose is eliminating outlier noise in data matrices, especially if a high percentage of the matrix points are contaminated with outlier noise.

Averaged over the different noise amplitudes and noise exposure percentages investigated, the WM method outperformed the standard median filtering procedure on the different data sets by 14–23% regarding data distance calculated with L_1 norm for eliminating non-zero mean noises. The version of the method for filtering zero mean noises, performed better by 14.3% on average against the conventional median filter.

Beyond general refinement and optimisation of the method, there is room for improvement particularly in more effective handling of the low noise exposure cases.

Funding Open access funding provided by University of Miskolc. None

Data availability Because of the data confidentiality, the experimental data is not published.

Code availability Because of the data confidentiality, the code is not published.

Declarations

Conflict of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Dobróka M, Gyulai Á, Ormos T, Csókás J, Dresen L (1991) *Joint inversion of seismic and geoelectric data recorded in an underground coal mine*. Geophysical Prospecting, 39 (5). ISSN 0016–8025:643–665. <https://doi.org/10.1111/j.1365-2478.1991.tb00334.x>
- Dobróka T (2021) *An MFV-based image processing filter and its application in seismic tomographic images*. Acta Geodaetica et Geophysica. <https://doi.org/10.1007/s40328-021-00351-7>
- Huang TS, Yang GJ, Tang GY (1979) A fast two-dimensional median filtering algorithm. IEEE Trans Acoust Speech Signal Process 27(1):13–18. <https://doi.org/10.1109/TASSP.1979.1163188>
- Scott DW (1979) On optimal and data-based histograms. Biometrika 66:605–610
- Scott DW (1992) Multivariate density estimation: theory, practice, and visualization. John Wiley & Sons, New York
- Steiner F (1991) *The Most Frequent Value. Introduction to Modern Conception Statistics, Budapest*. ISBN 978–9630556873
- Stone DC (1995) Application of median filtering to noisy data. Can J Chem 73(10):1573–1581. <https://doi.org/10.1139/v95-195>
- Szabó NP, Balogh GP, Stickel J (2018) Most frequent value-based factor analysis of direct-push logging data. Geophys Prospect. <https://doi.org/10.1111/1365-2478.12573>
- Szabó N, Balogh GP (2016) *Most frequent value based factor analysis of engineering geophysical sounding logs*. 78th EAGE Conference and Exhibition 2016, Houten, Holland: European Association of Geoscientists and Engineers (EAGE), Paper: Tu SBT4 12, 5 p. <https://doi.org/10.3997/2214-4609.201600796>
- Szűcs P, Zákányi B (2007) *Applying most frequent value (MFV) in hydrogeological modelling*. Műnőkeológia-Kőzetmechanika 161–174.
- Zhang J (2017) Most frequent value statistics and distribution of ^7Li abundance observations. Mon Not R Astron Soc 468(4):5014–5019. <https://doi.org/10.1093/mnras/stx627>