FAST OPTIMIZATION OF WEIGHTED VECTOR MEDIAN FILTERS

Yuzhong Shen and Kenneth E. Barner

University of Delaware
Department of Electrical and Computer Engineering
Newark, DE 19716, USA

ABSTRACT

In this paper, we analyze several previous optimization approaches for weighted vector median (WVM) filters and show their deficiencies. We then propose two fast adaptive WVM optimization algorithms. Proposed algorithm I computes the *optimal weight changes* at each iteration, and updates weights accordingly. Proposed algorithm II extends the results from weighted median optimization to the vector case by a generalization of an error metric. Both algorithms are fast and stable, and perform well under a wide variety of circumstances.

1. INTRODUCTION

Weighted vector median (WVM) filters were proposed by Viero *et al*. [1] for color image sequence filtering. The behavior of WVM filter can be controled by a set of weights assigned to the observation samples. To attain desired behavior and characteristics, the weights of WVM filters must be determined in an optimal manner.

Lucat *et al*. [2] derived the *optimal* weights under an L_p error criteria using a stochastic gradient method. Derivatives of the WVM filter output with respect to the weights are approximated by differences. Two derivative approximations were proposed: local approximation and mean output sensitivity. More recently, Lukac *et al*. [3] tried to extend the results from weighted median (WM) optimization to the vector case.

In this paper, we analyze the deficiencies of these previous algorithms. Specifically, the derivative approximations [2] are not appropriate, resulting in the instability of the steepest descent algorithm. The vector extension of fast LMA algorithm in [3] disregards the crucial direction information of vectors and is thus not justified. We propose two new WVM optimization algorithms. Proposed algorithm I computes the *optimal weight changes* at each iteration, and updates weights to reflect these changes. Proposed algorithm II extends the fast LMA algorithm for weighted median optimization to the vector case by a generalization of the error metric. Both algorithms are fast and stable, and perform well under a wide variety of circumstances.

2. RELATED WORK

For an observation window $\Omega = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N \in \mathbb{R}^m\}$ associated with a set of weights $\{w_1, w_2, \ldots, w_N\}$, the output of the WVM filter is defined as

$$\mathbf{y} = \arg\min_{\mathbf{x}\in\Omega} \sum_{i=1}^{N} w_i \left\|\mathbf{x} - \mathbf{x}_i\right\|_p, \qquad (1)$$

where $\|\cdot\|_p$ denotes the L_p norm. In this paper, only the L_1 norm is considered. The goal of the optimization is to find a set of weights minimizing the cost function $E[\|\mathbf{D}(n) - \mathbf{y}(n)\|_1]$, where $E[\cdot]$ is the expectation operator, $\mathbf{D}(n)$ the desired output, and $\mathbf{y}(n)$ the WVM output.

2.1. Optimization Based on Derivative Approximation

The optimal WVM weights can be solved using a stochastic gradient method,

$$w_i(n+1) = w_i(n) + \mu \operatorname{sgn}\left(\mathbf{D}(n) - \mathbf{y}(n)\right) \cdot \frac{\partial \mathbf{y}(n)}{\partial w_i}, \qquad (2)$$

for $i = 1, \ldots, N$ and μ is the iteration step size. As noted in [2], the derivative $\partial\mathbf{y}/\partial w_i$ is null for all i, thus rendering (2) useless. To overcome this inconvenience, Lucat *et al*. [2] proposed two derivative approximations: local approximation and mean output sensitivity. The local approximation is computed as

$$\frac{\partial \mathbf{y}(n)}{\partial w_i} = \lim_{\substack{|\delta w_i|\to 0 \\ \delta\mathbf{y}\neq\mathbf{0}}} \frac{\delta \mathbf{y}(n)}{\delta w_i}, \quad i = 1, \ldots, N. \qquad (3)$$

Assume $w_i^{j_1}$ is the smallest change of weight w_i in magnitude such that the WVM output changes from sample $\mathbf{x}_{j_0}$ to sample $\mathbf{x}_{j_1}$, then (3) can be written as

$$\frac{\partial \mathbf{y}(n)}{\partial w_i} = \frac{\mathbf{x}_{j_1} - \mathbf{x}_{j_0}}{\delta w_i^{j_1}}, \quad i = 1, \ldots, N. \qquad (4)$$

All the possible changes δw_i^j of weight w_i such that the WVM output switches from $\mathbf{x}_{j_0}$ to $\mathbf{x}_j$ can be solved by [2]

$$\delta w_i^j = \frac{d(\mathbf{x}_j) - d(\mathbf{x}_{j_0})}{\|\mathbf{x}_{j_0} - \mathbf{x}_i\|_1 - \|\mathbf{x}_j - \mathbf{x}_i\|_1}, \quad j = 1, \ldots, N, \quad (5)$$

where $d(\mathbf{x}_j) = \sum_{k=1}^{N} w_k \|\mathbf{x}_j - \mathbf{x}_k\|_1$. Then the index j_1 is found by

$$j_1 = \arg\min_j |\delta w_i^j|. \quad (6)$$

Substituting (6) into (4), the local approximation of derivatives is obtained.

Lucat *et al.* [2] also defined a *mean output sensitivity* to obtain a more *global* derivative approximation,

$$\frac{\partial \mathbf{y}(n)}{\partial w_i} = \frac{1}{|S|} \sum_{\mathbf{x}_j \in S} \frac{\mathbf{x}_j - \mathbf{x}_{j_0}}{\delta w_i^j}, \quad i = 1, \ldots, N, \quad (7)$$

where S denotes the entire set of potential output vectors involved when increasing or decreasing the weight w_i, and $|S|$ the cardinality of the set S.

2.2. Weighted Median Optimization Extension

The fast LMA algorithm [4] has been extensively used for the optimization of weighted median filter:

$$w_i(n+1) = w_i(n) + \mu(D(n) - y(n))\mathrm{sgn}(x_i(n) - y(n)), \quad (8)$$

where $x_i(n)$ are the inputs, $D(n)$ the desired output, and $y(n)$ the actual WM filter output. Lukac [3] *et al.* defined a signed difference measure for vectors,

$$S(\mathbf{a} - \mathbf{b}) = \begin{cases} +\|\mathbf{a} - \mathbf{b}\|_p, & \text{if } |\mathbf{a}| - |\mathbf{b}| > 0 \\ -\|\mathbf{a} - \mathbf{b}\|_p, & \text{if } |\mathbf{a}| - |\mathbf{b}| < 0. \end{cases} \quad (9)$$

Then they extended (8) to the vector case:

$$w_i(n+1) = w_i(n) + 2\mu S(\mathbf{D}(n) - \mathbf{y}(n))\mathrm{sgn}(S(\mathbf{x}_i(n) - \mathbf{y}(n))). \quad (10)$$

3. FAST WVM OPTIMIZATION

3.1. Analysis of Derivative Approximations

To analyze the derivative approximations (3) and (7), we utilize a simplified mathematical model. Two functions $J_C(w)$ and $J_D(w)$ are shown in Fig. 1, where $J_C(w)$ is a continuous function with a local minimum at w_0 and $J_D(w)$ is a piecewise constant function with a local minimum in the interval $[w_c, w_d]$. The steepest descent method is used to solve for these minimums.

The plot of $\left|\frac{\partial J_C(w)}{\partial w}\right|$ is shown in Fig. 2(a). It can be seen that $\left|\frac{\partial J_C(w)}{\partial w}\right|$ is continuous, and its value is 0 when $w = w_0$. When approaching w_0 from left or right, the value of

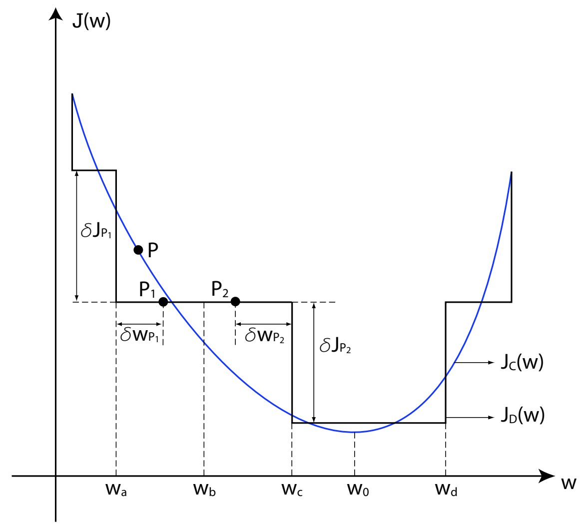


Fig. 1. Analysis of derivation approximation.

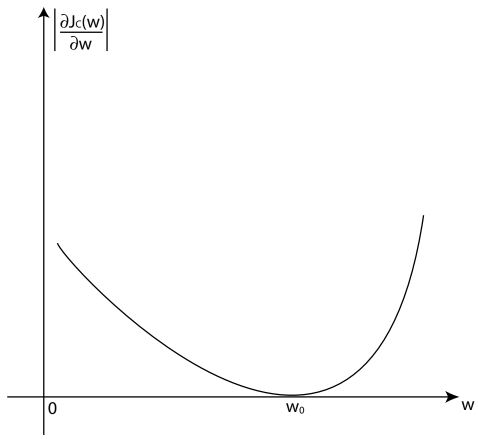
$\left|\frac{\partial J_C(w)}{\partial w}\right|$ is decreasing to 0 monotonically. These properties are necessary conditions required by the steepest descent algorithm to guarantee convergence at w_0.

Now consider $J_D(w)$ on the interval $[w_a, w_c]$, where $w_b = (w_a + w_c)/2$. Two points P_1 and P_2 are on $J_D(w)$, where $w_a < w_{P_1} < w_b$ and $w_b < w_{P_2} < w_c$. Based on the local approximation (3), the derivative at P_1 is computed as

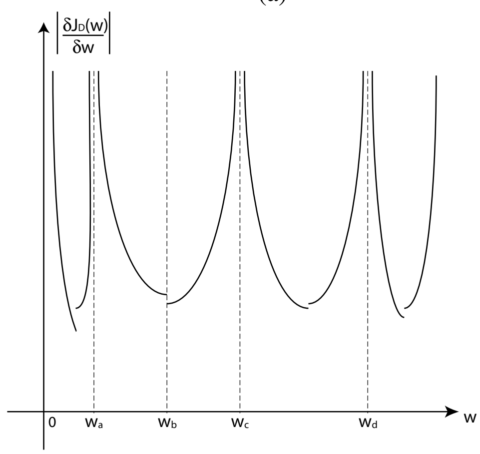
$$\frac{\partial J_D(w)}{\partial w} = \frac{\delta J_{P_1}}{\delta w_{P_1}}, \quad (11)$$

where δJ_{P_1} and δw_{P_1} are shown in Fig. 1. Note that δw_{P_1} is the smallest change of w in magnitude such that $J_D(w)$ changes its value. It is clear that $\left|\frac{\delta J_{P_1}}{\delta w_{P_1}}\right| \to \infty$ when $w_{P_1} \to w_a^+$, and $\left|\frac{\delta J_{P_1}}{\delta w_{P_1}}\right| \to \left|\frac{\delta J_{P_1}}{w_b - w_a}\right|$ when $w_{P_1} \to w_b^-$. In addition, $\left|\frac{\delta J_{P_1}}{\delta w_{P_1}}\right|$ is monotonically decreasing on $[w_a, w_b]$. Similarly, the derivative approximation $\left|\frac{\delta J_{P_2}}{\delta w_{P_2}}\right|$ increases from $\left|\frac{\delta J_{P_2}}{w_c - w_b}\right|$ to ∞ on $[w_b, w_c]$. Thus for the interval $[w_a, w_c]$ on which $J_D(w)$ is a constant, the approximation of $\left|\frac{\partial J_D(w)}{\partial w}\right|$ first decreases from ∞ to a minimum value, then increase to ∞ again. The magnitude of local derivative approximation of $J_D(w)$ is shown in Fig. 2(b). We can see that for each flat region of $J_D(w)$, the approximation of $\left|\frac{\partial J_D(w)}{\partial w}\right|$ first decreases from ∞ to a minimum value at the middle point, then increases to ∞ again. This unbounded and discontinuous shape of derivative approximation violates the convergence conditions required by the steepest descent method, resulting in erratic behavior of the optimization based on local approximation (3).

Since the derivative approximation (7) involves the computation of (4), which has been shown violating the conver-



(a)



(b)

Fig. 2. Magnitude of derivative approximation.

gence condition required by the steepest descent method, the optimization algorithm based on (7) does not produce satisfactory results either.

3.2. Analysis of WM Optimization Extension

Consider two samples $\mathbf{x}_1$ and $\mathbf{x}_2$ with same magnitudes, i.e., $|\mathbf{x}_1| = |\mathbf{x}_2|$. Based on the optimization scheme (10), the weights w_1 and w_2 corresponding to $\mathbf{x}_1$ and $\mathbf{x}_2$ will be changed by an equal amount regardless of their directions. For example, if $\mathbf{x}_1$ and $\mathbf{x}_2$ represent two color pixels $(255, 0, 0)$ and $(0, 255, 0)$, their weights will be equally changed although they represent distinct colors. Thus this optimization approach [3] is not justified since it disregards the crucial direction information of vectors.

3.3. Proposed Algorithm I: Fast Greedy Optimization

To address the problems in the derivative approximations (3) and (7), we propose the following optimization procedures that do not involve derivative computations:

1. Set initial values of filter weights.
2. Compute the cost of each input sample as
$$e_i = \|\mathbf{x}_i(n) - \mathbf{D}(n)\|_1\,, \quad i = 1, \ldots, N. \tag{12}$$
3. Find the sample $\mathbf{x}_{e_{min}}$ that has the minimum cost,
$$\mathbf{x}_{e_{min}} = \arg\min_{\mathbf{x}\in\Omega} e_i = \arg\min_{\mathbf{x}\in\Omega} \|\mathbf{x} - \mathbf{D}(n)\|_1\,. \tag{13}$$
4. If $\mathbf{x}_{e_{min}}$ is the current filter output, set $\Delta w_i = 0$. Otherwise, compute the necessary weight change Δw_i (5) so that $\mathbf{x}_{e_{min}}$ becomes the filter output,
$$\Delta w_i = \frac{d(\mathbf{x}_{e_{min}}) - d(\mathbf{x}_{j_0})}{\|\mathbf{x}_{j_0} - \mathbf{x}_i\|_1 - \|\mathbf{x}_{e_{min}} - \mathbf{x}_i\|_1}, \tag{14}$$
where $d(\cdot)$ and $\mathbf{x}_{j_0}$ are defined as before.
5. Update the filter weights
$$w_i(n+1) = w_i(n) + \mu\Delta w_i(n),\ i = 1, \ldots, N, \tag{15}$$
where μ is the iteration step size.

Since the goal of optimization is to find a set of weights that minimize the cost function, we search for the sample $\mathbf{x}_{e_{min}}$ with the minimum cost at each iteration. If $\mathbf{x}_{e_{min}}$ is the current filter output, the WVM weights remain unchanged. Otherwise, we compute the necessary weight changes such that $\mathbf{x}_{e_{min}}$ becomes the output and update the weights to reflect these changes. The weight change Δw_i (14) may be too large under certain circumstances, and in such cases, Δw_i is clamped such that $|\Delta w_i| < T$, where T is a threshold.

3.4. Proposed Algorithm II: Fast LMA Vector Extension

The fast LMA algorithm (8) for the WM optimization regulates the filter weights based on the relations between the desired output, the actual output, and the inputs. We can rewrite (8) as

$$\begin{aligned} w_i(n+1) &= w_i(n) - \mu(y(n) - D(n))\mathrm{sgn}((x_i(n) - D(n)) \\ &\qquad - (\hat{D}(n) - D(n))) \\ &= w_i(n) - \mu\, e(y(n))\mathrm{sgn}(e(x_i(n)) - e(y(n))), \end{aligned} \tag{16}$$

where $e(y(n)) = y(n) - D(n)$ and $e(x_i(n)) = x_i(n) - D(n)$. To extend (16) to vector signals, we define $e(\mathbf{y}(n)) = \mathrm{sgn}\left(\|\mathbf{y}(n)\|_1 - \|\mathbf{D}(n)\|_1\right) \cdot \|\mathbf{y}(n) - \mathbf{D}(n)\|_1$ and $e(\mathbf{x}_i(n)) = \mathrm{sgn}\left(\|\mathbf{x}_i(n)\|_1 - \|\mathbf{D}(n)\|_1\right) \cdot \|\mathbf{x}_i(n) - \mathbf{D}(n)\|_1$, and obtain the following WVM optimization algorithm :

$$\begin{aligned} w_i(n+1) &= w_i(n) - \mu\, e(\mathbf{y}(n))\mathrm{sgn}\left(e(\mathbf{x}_i(n)) - e(\mathbf{y}(n))\right) \\ &= w_i(n) - \mu\, e_o \mathrm{sgn}(e_i - e_o), \quad i = 1, \ldots, N \end{aligned} \tag{17}$$

where $e_o = e(\mathbf{y}(n))$ and $e_i = e(\mathbf{x}_i(n))$. From the iteration (17), when $\|\mathbf{y}(n)\|_1 > \|\mathbf{D}(n)\|_1$, the weight w_i is increased if $\|\mathbf{x}_i(n)\|_1 < \|\mathbf{D}(n)\|_1$ or $\|\mathbf{x}_i(n) - \mathbf{D}(n)\|_1 < \|\mathbf{y}(n) - \mathbf{D}(n)\|_1$, and is decreased otherwise. Similar analysis can be performed for the case of $\|\mathbf{y}(n)\|_1 < \|\mathbf{D}(n)\|_1$.

4. RESULTS AND DISCUSSION

As a typical multivariate application, we optimize the WVM filter for color image denoising. The proposed algorithms have been used to compute the optimal WVM weights for denoising various color images, and they perform remarkably well under a wide variety of circumstances.

Due to space limit, here we only present one example. Figure 3 shows the optimization results for a color image (not shown) from Lucat's method [2] and the proposed algorithm I and II. The optimal weights for a single realization are represented by 5×5 meshes in Figures 3(a), (c), and (e). The corresponding weight variances for multiple realizations of the same noise distribution are shown in Figures 3(b), (d), and (f). It can be seen that the proposed algorithms are extremely stable with nearly zero variances for different realizations of the same noise distribution, while Lucat's algorithm is erratic. The averages and variances of the mean absolute error (MAE) of WVM filtering with the obtained optimal weights for multiple realizations are listed in Table 1, which further demonstrates the superior performance of the proposed methods. Furthermore, the proposed methods are significantly faster than the Lucat's method [2], with computation time of 130 seconds vs 500 seconds for a 482×467 color image in this example. The size of the WVM filter used is 5×5.

Compared to the algorithm I, the proposed algorithm II is slightly faster but more sensitive to the step size μ. However, they have similar performance in terms of optimization results and numerical stability. Future work includes more theoretical study of both algorithms, such as convergence condition and rate, and their adoption to other applications involving multivariate data.

Table 1. MAE Average and Variance

Optimization Method	MAE	
	Average	Variance
Lucat's algorithm	6.62	5.34
Proposed Algorithm I	4.08	2.61×10^{-5}
Proposed algorithm II	4.06	1.07×10^{-4}

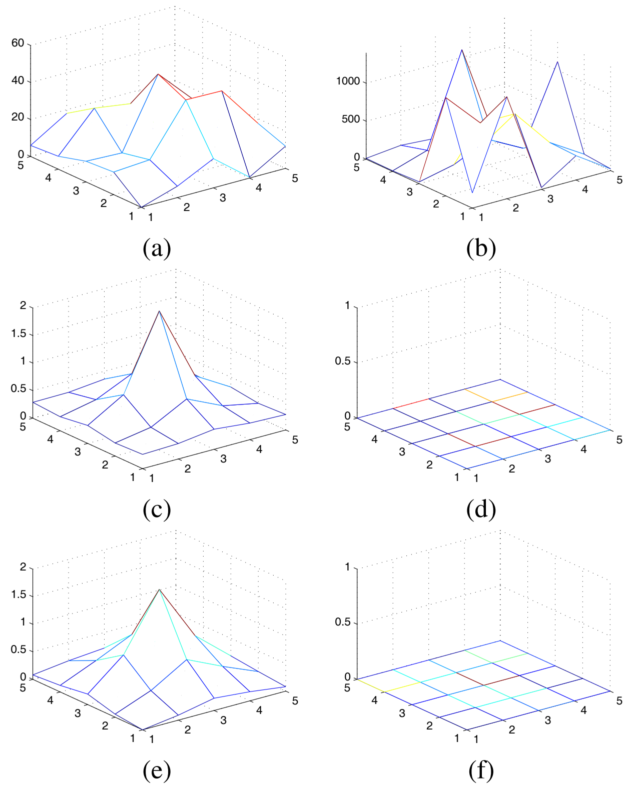


Fig. 3. Optimization results for color image denoising. Plots (a), (c), and (e) show the optimal weights obtained from Lucat's algorithm and the proposed algorithm I and II for a single realization, and plots (b), (d), and (f) show the corresponding weight variances for multiple realizations of the same noise distribution. The filter window size is 5×5.

5. REFERENCES

[1] Timo Viero, Kai Öistämö, and Yrjö Neuvo, "Three-dimensional median-related filters for color image sequence filtering," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 4, no. 2, pp. 129–142, April 1994.

[2] Laurent Lucat, Pierre Siohan, and Dominique Barba, "Adaptive and global optimization methods for weighted vector median filters," *Signal Processing: Image Communications*, vol. 17, no. 7, pp. 509–524, 2002.

[3] R. Lukac, K. N. Plataniotis, B. Smolka, and A. N. Venetsanopoulos, "Weighted vector median optimization," in *4th EURASIP Conference focused on Video/Image Processing and Multimedia Communications*, Zagreb, Croatia, 2003, pp. 227–232.

[4] Lin Yin, Ruikang Yang, Moncef Gabbouj, and Yrjö Neuvo, "Weighted median filters: A tutorial," *IEEE Transactions on Circuits and Systems -II: Analog and Digital Signal Processing*, vol. 43, no. 3, pp. 157–192, March 1996.