MEDIAN FILTER WITH ABSOLUTE VALUE NORM SPATIAL REGULARIZATION

Nilanjan Ray
Department of Computing Science
University of Alberta, Canada
nray1@ualberta.ca

ABSTRACT
We provide a novel formulation for computing median filter with spatial regularization as minimizing a cost function composed of absolute value norms. We turn this cost minimization into an equivalent linear programming (LP) and solve its dual LP as a minimum cost flow (MCF) problem. The MCF is solved over a graph constructed for an input image, and the primal LP solution is retrieved as the filtered image. For solving the MCF, we utilize an efficient network simplex algorithm. Numerical results show that the proposed median filter with a spatial regularization term outperforms median filters and a decision theoretic filter for impulse noise removal.

1. INTRODUCTION
Median filter is one of the earliest non-linear filters in signal and image processing. Given its simple computations and effectiveness in removing outliers, median filter is still very much relevant in various tasks ranging from impulse noise removal [1] to high quality optical flow computations [2]. Other than handling of outliers, median filter also possesses several important properties: edge preservation, self-duality, invariance to increasing/decreasing illuminance change, etc. [3]. Over the years, various improvements on basic median filters have been made. Adaptive median filters [4], multistate median filter [5], median filter based on region homogeneity [6], and so on have emerged.

In this paper, we take a novel direction towards improving median filters by adding a spatial regularization to median data fidelity. At first, it might seem superfluous, because median filter already possesses spatial regularization as a result of its sliding window operation. However, we illustrate here that adding an extra spatial regularization enhances its effectiveness manifold. We interpret a median filter as optimizing an absolute value (also known as $L_1$) norm cost minimization [7]. Next, we add another $L_1$ norm spatial term to the median filter cost. We propose a novel minimization method for this combined cost function by means of an efficient minimum cost flow (MCF) algorithm [8]. The use of $L_1$ norm is justified, because among all $L_p$ norms, $p = 1$ is the smallest number that yields a convex cost with the most robust penalty toward outliers (e.g., edge preservation).

In the image denoising context, $L_1$ penalty has been used before as a data fidelity term [9]. A close relative of $L_1$ penalty, the total variation (TV) norm, has been used as regularization in Rudin-Osher-Fatemi (ROF) model of image smoothing [10]. Fast computational techniques have been proposed for ROF in Chambolle’s work [11]. While ROF model uses $L_2$ norm for the data fidelity term, $L_1$ norm has been utilized for the data and the regularization terms with a fast parametric max-flow implementation for edge preserving image smoothing [12].

Our work differs from these aforementioned image restoration work in at least two counts. First, our data fidelity term is that of a median filter, which maintains multiple hypotheses. Using order statistics, the data term chooses a value from among several values (multiple hypotheses). Secondly, while most other methods avoided $L_1$ regularization, principally because of the non-smooth nature of the $L_1$ penalty, we have utilized basic techniques of linear programming (LP) to handle this non-smoothness. Our treatment of reducing the optimization problem to an equivalent LP is novel.

Our computational approach leads to a simple graph construction for which a network simplex implementation [8] is utilized from Matlab/CPLEX optimization package. Results show the effectiveness of our method in removing speckle noise and high density salt and pepper noise.

2. PROPOSED FORMULATION
Let $I$ denote an input gray scale image. Suppose, we want to compute a weighted median filter output $u$ for the input $I$. It is well known that weighted median filter is the solution of the following minimization [7]:

$$\min_u \sum_{i \in V} \sum_{l \in N_i} w_{il} |u_i - I_l|,$$

where $V$ is the set of pixel locations in the image, $N_i$ is a set consisting of the pixel $i$ and the neighboring pixels of $i$, and $w_{il}$ are non-negative weights.

Supported by NSERC Discovery Grant
Fig. 1. The image here is a 2-by-2 image for illustrative purpose, with pixel locations \( V = \{1,2,3,4\} \) and 4-neighboring edges \( E = \{(1,2),(1,3),(2,4),(3,4)\} \) appearing as dark edges. A special node \( t \) is introduced along with the associated data edges (light edges) to map the MCF (4) to this graph.

In this paper we consider a median filter with an \( L_1 \) norm spatial regularization term as:

\[
\min_u \sum_{i \in V} \sum_{l \in N_i} w_{il} |u_i - I_l| + \sum_{(i,j) \in E} \lambda_{ij} |u_i - u_j|, \quad (2)
\]

where \( E \) is the set of pixel neighbors for the spatial regularization, such as the set of 4- or 8-neighborhood pixels. However, \( E \) can be any suitable set of connections in the image pixel grid. Also note that this neighborhood structure \( E \) can be different from the neighborhood structure \( N_i \). The non-negative parameters \( \lambda_{ij} \) balance the weight between the data fidelity (the median) and the smoothness terms. Although convex, it is difficult to directly minimize (2), because of the non-differentiable nature of the \( L_1 \) norm. We can, however, reformulate (2) as the following LP:

\[
\begin{align*}
\min & \sum_{i \in V} \sum_{l \in N_i} w_{il} p_{il} + \sum_{(i,j) \in E} \lambda_{ij} q_{ij}, \\
\text{s.t.} & \quad p_{il} \geq u_i - I_l, \quad p_{il} \geq I_l - u_i, \\
& \quad q_{ij} \geq u_i - u_j, \quad q_{ij} \geq u_j - u_i.
\end{align*}
\]

Using a standard LP package to solve (3) is quite inefficient and sometimes not practicable for even a moderately sized image. To find out a speedy solution to (3) we use the principle of duality. The dual LP of (3) takes the following form:

\[
\begin{align*}
\min & \sum_{l \in N_i} x_{il} I_l, \\
\text{s.t.} & \quad -w_{il} \leq x_{il} \leq w_{il}, \\
& \quad -\lambda_{ij} \leq y_{ij} \leq \lambda_{ij}, \\
& \quad \sum_{l \in N_i} x_{il} + \sum_{j: (i,j) \in E} y_{ij} - \sum_{j: (j,i) \in E} y_{ji} = 0, \quad \forall i \in V.
\end{align*}
\]

The LP (4) is readily recognized as the LP associated with a minimum cost flow (MCF) problem [8] for which we construct a graph shown in Fig. 1. For the sake of producing a relatively clear picture here, we consider a 2-by-2 tiny image with 4-connectivity edge set \( E \). The neighborhood \( N_i \) is assumed to have only two pixel locations, once again for the sake of an uncluttered illustration. We notice that the graph contains four pixels locations (1, 2, 3, and 4) along with a special vertex \( t \). The data edges appear as \( x_{il} \) and the regularization edges as \( y_{ij} \). From each pixel node \( i \) there are two data edges linking \( t \), because, here \( N_i \) contains \( i \) and only one other neighbor of \( i \). There is no cost on the regularization edges, while the cost of sending unit flow along a data edge \( x_{il} \) is \( I_l \). The data edge capacities vary between \(-w_{il}\) and \(+w_{il}\), while those of the regularization edges vary between \(-\lambda_{ij}\) and \(+\lambda_{ij}\).

Notice that the flow balance equation for the node \( t \) can be derived by summing all the flow balance equations in (4):

\[
\sum_{i \in V} \sum_{l \in N_i} x_{il} = 0. \quad (5)
\]

Thus, the MCF given by equation (4) is essentially a minimum cost circulation. An important property for the primal solution \( u \) can be derived from the graph in Fig. 1 as follows. An MCF can be solved using a negative cycle canceling algorithm that leaves us to retrieve the primal LP solution by solving a shortest path problem on the residual graph [8] starting at \( t \). Because, there are no costs on the edge links \( E \), any negative cost cycle must include the node \( t \). After the termination of a negative cycle canceling algorithm, no negative cost cycles exist in the residual graph. So, from the node \( t \), the cost of the shortest path (primary LP solution) to any node \( i \) must be only a cost appearing on a data edge, i.e., an input image intensity value. In other words, the solution of our spatially regularized median filter will contain only input image intensity values, like the output of a weighted median filter.

3. RESULTS

We compare the proposed median filter with ordinary median filters and a decision-based filter (DBF) for impulse noise removal [13]. DBF is the latest and one of the fastest methods for high density impulse noise removal. DBF mimics a median filter, which discards corrupted pixel values in median computation. In Fig. 2(a) and (b) we show the original image and its noisy version. In this case, we have randomly chosen approximately 50% of the pixels of the original image and corrupted their values with salt and pepper type impulse noise. Fig. 2(c) shows the result of applying a 3-by-3 median filter on the original image. In Fig. 2(d) we have illustrated the result of applying our proposed spatially regularized median filter. We have considered a 3-by-3 neighborhood for both \( N_i \) and \( E \). We chose unity for all the regularization weights in (2): \( \lambda_{ij} = 1 \). The data fidelity weights are set as follows:

\[
w_{il} = \begin{cases} w, & \text{if } 0 < I_l < 255; \\
0, & \text{otherwise.} \end{cases}
\]

1438
In producing Fig. 2(d), we chose $w = 100$. Notice that (6) uses a simple way to detect salt and pepper noise for a 256-level gray scale image $I$. For noisy pixels, data fidelity weights are set to 0.

In a similar experiment, we add 90% salt and pepper noise to the Cameraman image and apply a 5x5 median filter, the DBF [13] and the proposed filter for denoising. The results are shown in Fig. 3. In this case again, we chose all weights and neighborhoods for the proposed filter as before. From these visual comparisons, we can clearly see that noise removal capability of an ordinary median filter is significantly enhanced by adding an $L_1$ regularization. Our method also compares favorably against the DBF method.

Table 1 provides a comparison of peak signal-to-noise ratio (PSNR) values for different methods. We varied the % of salt and pepper noise from 60% to 95% for the Cameraman image. In the proposed method, we used a 3-by-3 neighborhood size for both $N_i$ and $E$, as before. We also set $\lambda_{ij} = 1$ in (2) and $w = 100$ in (6) as has been done in Fig. 2(d) and 3(d). It is observed that the spatially regularized median filter has cleaned the Cameraman image significantly better than the other two methods. Even at high level of noise, our method was able to preserve edges better. The Cameraman image is 256x256 in size. Average computation time for the decision-based filter (Matlab implementation) is 0.8 second, while our method (Matlab implementation with call to CPLEX library) takes about 1.5 seconds on a Linux machine with 4 GB RAM, 3 GHz Intel Core 2 processor.

To see the effect of the user input parameter $w$ in (6) on the noise removal capability of the proposed filter, we have produced a $w$ vs. PSNR plot in Fig. 4. In this case the Cameraman image was corrupted with 75% salt and pepper noise. We notice that beyond a certain value of the edge weight $w$, 

![Fig. 2](image1)

Fig. 2. (a) Cameraman image; (b) 50% salt and pepper noise added; (c) 3-by-3 median filter; (d) 3-by-3 median filter with 3-by-3 spatial regularization.

![Fig. 3](image2)

Fig. 3. (a) 90% salt and pepper noise added to Cameraman image; (b) 5-by-5 median filter; (c) DBF method; (d) Proposed method.

![Fig. 4](image3)

Fig. 4. $w$ vs. PNSR plot for the proposed method. Noise level is 75%.

<table>
<thead>
<tr>
<th>Method \ Noise</th>
<th>60%</th>
<th>70%</th>
<th>80%</th>
<th>90%</th>
<th>95%</th>
</tr>
</thead>
<tbody>
<tr>
<td>5x5 Median</td>
<td>16.09</td>
<td>12.74</td>
<td>9.5</td>
<td>6.89</td>
<td>5.94</td>
</tr>
<tr>
<td>DBF</td>
<td>21.38</td>
<td>20.98</td>
<td>20.33</td>
<td>19.37</td>
<td>18.29</td>
</tr>
<tr>
<td>Proposed</td>
<td>23.65</td>
<td>23.07</td>
<td>22.08</td>
<td>20.82</td>
<td>19.84</td>
</tr>
</tbody>
</table>

Table 1. Comparison of PSNR values for denoising Cameraman image corrupted with varying degrees of noise.
the PSNR does not change. The plot also justifies our choice of parameter value \( w = 100 \) for earlier experiments. This plot reveals that setting the weight to a large value does not hurt the performance much. This can be seen as an advantage of the proposed method.

Fig. 5(a) shows the Cameraman image corrupted with speckle noise. Fig. 5(b) and (c) show results of median filters. Fig. 5(d) shows the result of the proposed filter. We have used a 3x3 neighborhood for both \( N_i \) and \( E \). The data weights are all set as: \( w_{il} = 0.5 \). Edge weights are set to depend on image intensities: \( \lambda_{ij} = \exp(-(I_i - I_j)^2/50^2) \). Note that the spatially regularized median filter has worked significantly better than ordinary median filters.

4. SUMMARY AND FUTURE WORK

In this paper, we propose a novel modification to median filter by adding an absolute value norm spatial regularization to an absolute value norm data fidelity term. We optimize the proposed convex, non-smooth cost function by converting it to an equivalent, novel MCF problem. We solve the MCF problem by a network simplex algorithm. We have illustrated the efficacy of our spatially regularized median filter with respect to high-density impulse noise removal. In the future, we would like to replace the network simplex method with a real-time parametric max-flow implementation much like [12].

5. REFERENCES