Making Joint-Histogram-Based Weighted Median Filter Much Faster*  

Hanhoon PARK[a), Member

SUMMARY In this letter, we propose a simple framework for accelerating a state-of-the-art histogram-based weighted median filter at no expense. It is based on a process of determining the filter processing direction. The determination is achieved by measuring the local feature variation of input images. Through experiments with natural images, it is verified that, depending on input images, the filtering speed can be substantially increased by changing the filtering direction.

key words: weighted median filter, joint-histogram, color or intensity variation, Sobel operator

1. Introduction

A number of types of spatial and spectral filters have been developed, and these played an important role in the field of computer vision. Among them, weighted median filter (WMF) has been popularly utilized for filtering images while preserving edges. Furthermore, since it is mathematically connected to global optimization [1], it can be applied to various computer vision applications such as stereo matching [2], optical flow estimation [3], and image denoising [1].

However, since WMF is based on sorting the pixels within a local window to find where the median is, its computational complexity is high. For an unweighted median filter, which can be considered as a special case of WMF, several solutions have been proposed [4], [5]. They initially compute a histogram of pixel values in a local window and quickly update it in the next local window using the prior that histograms of adjacent local windows are almost the same. Then, the median can be found by counting elements in each bin. Although the approaches enable unweighted median filter to work in constant time, they are not applicable to WMF where weights vary according to feature distance [3]. This is because, due to the varying weights, histograms of adjacent local windows are no longer similar in WMF.

Recently, Ma et al. proposed a constant time WMF for stereo matching [2], which is based on 2D box filtering and integral imaging [6]. However, it required that input images contain only a limited number of intensity levels and the weights be pre-defined. Therefore, most recently, Zhang et al. proposed a general fast WMF that can handle higher precision input images and arbitrary weight assignment [3]. Their WMF was extremely faster than the brute-force WMF and even the constant time WMF [2].

In this letter, we propose a method for further accelerating Zhang et al.’s state-of-the-art WMF with no expense. The proposed method efficiently analyzes input image statistics and determines the direction of local window shift. Depending on input images, Zhang et al.’s WMF can be much faster by using the proposed method.

The remainder of this letter is organized as follows. In Sect. 2, we briefly describe Zhang et al.’s WMF. In Sect. 3, we detail the proposed method for accelerating the Zhang et al.’s WMF. Then, experimental results and conclusion are given in Sect. 4 and 5, respectively.

2. Joint-Histogram-Based WMF [3]

To accelerate the brute-forth WMF, Zhang et al.’s proposed three strategies: joint-histogram, median tracking, and necklace table.

First, the joint-histogram is 2D histogram which includes the counts not only for pixel values but also for their features, and can quickly regenerate weights every time the images are shifted.

---

*This study was funded by KARI.
a) E-mail: hanhoon_park@pknu.ac.kr  
DOI: 10.1587/transinf.2014EDL8144

Fig. 1 Horizontal and vertical scanline order.

Fig. 2 Image having significantly different median differences in horizontal and vertical scanline direction.
window shift. Therefore, weighted median filtering can be quickly done by two steps: median finding from the joint-histogram, local window shift and joint-histogram update. The steps are similar to those of the histogram-based approaches for unweighted median filter.

Second, to shorten the time taken for the median find-
Fig. 5  Images having similar median differences in both horizontal and vertical scanline direction.
Left: original image, middle: filtering result using color feature, right: filtering result using intensity feature.

ing step, Zhang et al. used the color and feature consistency of natural images: the difference between the weighted medians in adjacent local windows is around 7-8 for 8-bit grayscale images. With the prior, they introduced the concept of cut point, which indicates the location that the difference between the sum of weights at its left side in joint-histogram and that in its right side becomes the smallest positive. Then, given a cut point in a local window, they could find the new weighted median in the adjacent local window from a few shifts of the cut point. For this process, they also used a 1D table called balance counting box which records the difference between the two sides of the cut point.

Third, to shorten the time taken for traversing joint-histogram, Zhang et al. used the data sparsity in joint-histogram and balance counting box: 90%-98% bins are empty. That is, they introduced an effective data structure, called necklace table, which is an unordered circular double-linked list.

With the three strategies, the brute-force WMF could be 100+ times faster.

3. Proposed Method

In this letter, we focus on the point that Zhang et al. overlooked in their processes. That is, the Zhang et al.’s WMF processes the input image only in a horizontal scanline order (see Fig. 1 for two different scanline orders). However, their process could be much faster, simply by changing the processing direction. Notice that changing the processing direction does not influence the WMF results.

As mentioned in Sect. 2, Zhang et al. exploited the observation that the difference between the weighted medians in adjacent local windows is around 7-8 for 8-bit grayscale images. However, since the difference was computed only in a horizontal scanline order, there is a redundancy in the median tracking step. Depending on input images, the difference can be smaller in another computation direction. This means that the median tracking step can be much faster by
changing the direction of median tracking. For example, whereas the image of Fig. 2 has the median difference of 5 in the horizontal scanline direction, the difference is close to 0 in the vertical scanline direction. Therefore, the median tracking had better be done in the vertical scanline direction.

Of course, the image is not natural but there are a number of natural images that have largely different median differences in horizontal and vertical scanline direction. Such images will be shown later.

The main problem is how to compute which direction is better. As explained above, the direction where the median difference is smaller will be better. However, to determine the direction, we cannot compute all the medians in both directions. We know that the median difference is closely related to local feature consistency of natural images and thus attempt to use a simple and fast method that can measure local feature variation. In this letter, we use the 1D Sobel operator, $S_H = [10-1]$, $S_V = [10-1]^T$:

If $\text{var}_H < \text{var}_V$, horizontal WMF, 
If $\text{var}_H > \text{var}_V$, vertical WMF,

where $\text{var}_H = \text{mean}(S_H*I)$, $\text{var}_V = \text{mean}(S_V*I)$.

Here, $I$ is an input image and $\text{mean()}$ is the average function.

### 4. Experimental Results and Discussion

As explained in Sect. 3, our WMF is exactly the same as Zhang’s et al.’s WMF except for the process of determining the filter processing direction. Therefore, we will do edge-aware image filtering using the two WMFs with two different features (color and intensity) and show how much the additional process can improve the speed of Zhang et al.’s WMF. As mentioned before, the image results by the two WMFs are not different and thus not considered here. In the evaluation, the local window size was fixed to $10 \times 10$ and the Gaussian weight function with $\sigma = 25.5$ was used [3]. The experiments were conducted on a PC with an Intel i7 3.4GHz CPU, 8GB memory, and Windows 7 32bit OS.

### Table 1: Processing time [ms].

<table>
<thead>
<tr>
<th></th>
<th>Color feature used</th>
<th></th>
<th>Intensity feature used</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Horizontal*</td>
<td>Vertical</td>
<td>Horizontal*</td>
<td>Vertical</td>
</tr>
<tr>
<td>Fig. 3 (a)</td>
<td>549</td>
<td>692</td>
<td>584</td>
<td>690</td>
</tr>
<tr>
<td>Fig. 3 (b)</td>
<td>973</td>
<td>1177</td>
<td>769</td>
<td>934</td>
</tr>
<tr>
<td>Fig. 4 (a)</td>
<td>1423</td>
<td>979</td>
<td>1372</td>
<td>938</td>
</tr>
<tr>
<td>Fig. 4 (b)</td>
<td>1600</td>
<td>1152</td>
<td>1475</td>
<td>1067</td>
</tr>
<tr>
<td>Fig. 5 (a)</td>
<td>957</td>
<td>1000</td>
<td>786</td>
<td>827</td>
</tr>
<tr>
<td>Fig. 5 (b)</td>
<td>1333</td>
<td>1277</td>
<td>1017</td>
<td>961</td>
</tr>
<tr>
<td>Fig. 5 (c)</td>
<td>1460</td>
<td>1386</td>
<td>1200</td>
<td>1145</td>
</tr>
</tbody>
</table>

* All the results in the horizontal scanline order can be considered as those obtained by the Zhang et al.’s WMF.

C++ code for Zhang et al.’s WMF was provided by the authors [3]. Our algorithms were implemented using C++ and OpenCV library [7]. For experiments, a number of images, including synthetic images, those from www.flickr.com, and those used in [3], were used. A partial results are given here.

In results, we could divided the images used in experiments into three groups. First, the images in Fig. 3 had smaller feature (color or intensity in our experiments) variation in the horizontal direction and thus the filter processing in the horizontal scanline order was faster (the time was saved by about 19% for color feature and about 17% for intensity feature in Table 1). Second, the images in Fig. 4 had smaller feature variation in the vertical direction and thus the filter processing in the vertical scanline order was faster (the time was saved by about 30% for color feature and about 30% for intensity feature in Table 1). Third, the images in Fig. 5 had similar feature variation in both the horizontal and vertical direction and thus the filter processing in each scanline order had little difference (the time was similar in Table 1). Of course, it was still true that the filter processing in the direction having smaller feature variation is faster. As a result, for the first group, use of the Zhang et al.’s WMF is not a concern. Also it may not be a big concern for the third group. However, for the second group, it is a big concern and the proposed method is necessary.

Figure 5 (c) looks like having smaller feature variation in the vertical direction at a glance. However, the image actually had large color and intensity variation in the vertical
direction. Equation (1) correctly computed the variation. It indicates that it is difficult to determine the filter processing direction without the actual computation of the variation using Eq. (1).

The time taken for determining the filter processing direction using Eq. (1) was around 2-3 ms for grayscale features and thus ignorable when compared with the time saved by using the process.

In addition, we analyzed how the proposed method accelerates the Zhang et al.’s WMF on a natural image dataset (BSDS500 [8]). The images were $481 \times 321$ or $321 \times 481$, the average filtering time was 404 ms (for color feature) and 361 ms (for intensity feature) without the proposed method. However, by using the proposed method, maximum 20% and average 6% of the time could be saved for both features (only the images that have smaller median differences in the vertical scanline direction were used, which were about 37% of the dataset images). In a number of images, the difference between the horizontal and vertical variation was not so large and the overall improvement was less impressive unlike expected. Figure 6 shows a part of the dataset images.

5. Conclusion

We proposed an acceleration method for the joint-histogram-based WMF. The method analyzed the local feature variation of input images in both the horizontal and vertical direction using the 1D Sobel operator. Then, according to the analysis, it determined the computationally efficient filter processing direction, which made the filter speed much higher with no expense (Depending on input images, the speed became about 30% higher).

References