LA-UR-13-25331
Approved for public release; distribution is unlimited.

Title: DP: a Fast Median Filter Approximation

Author(s): Marcus, Ryan C.
Ward, William C.

Intended for: Web

Issued: 2013-07-23 (Rev.2)

DP: a Fast Median Filter Approximation

Ryan Marcus
AET-6
rmarcus@lanl.gov

William Ward
AET-6
ww@lanl.gov

July 16, 2013

Abstract

We present a new non-discrete algorithm that quickly approximates a median filter. This new algorithm proves to be faster than our implementations of many other fast median filter algorithms.

1 Introduction

Median filters can greatly reduce noise in an image, but are computationally expensive [10]. Fast median filter algorithms are required for batch image processing and real-time filtering.

Many have shown that fast median filter algorithms can be constructed through histogram manipulation [5, 11, 9]. These algorithms assume that data can be discretized into histogram bins, which is true for standard 8-bit images. However, many applications, such as computed tomography, utilize floating-point data [6]. Others have shown that efficient sorting mechanisms for small sets or statistical techniques can lead to fast results of varying accuracy [3, 7].

We present a median filter approximation algorithm that does not require data to be discretized. This algorithm performs favorably compared to other approximation algorithms, and has known error bounds.

2 Algorithm

Narendra demonstrated that applying a one-dimensional median filter across the rows and columns of an image can quickly produce accurate results [8]. However, giving special attention to various properties of the median function itself can yield an even faster algorithm. By combining the techniques of Narendra [8] and Huang [5], one can achieve faster results by exploiting data overlap.

The new algorithm, named DP, is outlined in algorithm 1 for a 3x3 ($W = 3$) median filter. Note that $I_{i,j}$ refers to the pixel in the ith row and jth column, with $I_{0,0}$ representing the upper left hand corner of the image.

DP is similar to Huang in that DP moves a kernel[1] across an image by row from left to right and preserves data between each kernel. Unlike Huang, DP does not maintain a histogram, but rather the median values of $w - 1$ columns. When progressing the kernel by one column, DP discards the median value of the previous leftmost column and calculates the median value of the new rightmost column. The median value of the kernel is then approximated by taking the median value of the column medians.

Table 1a represents the initialization step for each row in which the median of the first and second columns are calculated and stored into $col1$ and $col2$, respectively. This step occurs once at the beginning of every row. Table 1b shows how the first kernel is fully calculated by finding the median of the third column, stored into $col3$, and then finding the median of the medians (which is the median of the set $\{col1, col2, col3\}$). Table 1c shows how the kernel moves over to the next column: the value of $col2$ is placed into $col1$, the value of $col3$ is placed into $col2$, and the median of the new rightmost column is stored into $col3$. The median value of the kernel represented in table 1c is once again calculated by finding the median of the set $\{col1, col2, col3\}$.

While DP is not guaranteed to find the exact median of a given kernel, it is guaranteed to find a value near the median. The exact accuracy of DP is discussed and proven in the next section.

3 Accuracy

The accuracy of DP depends upon the size of the kernel used for filtering. We assume a WxW square. While the DP algorithm will work with any

[1] Many sources use the term "window" instead of "kernel."

Algorithm 1 The DP algorithm for median filter calculations

function DP(Image I, Image I')
  for $i = 1 \rightarrow$I.$HEIGHT - 1$ **do**
    $col1 \leftarrow$ `median`$([I_{i-1,0}, I_{i,0}, I_{i+1,0}])$
    $col2 \leftarrow$ `median`$([I_{i-1,1}, I_{i,1}, I_{i+1,1}])$
    for $j = 1 \rightarrow$I.$WIDTH - 1$ **do**
      $col3 \leftarrow$ `median`$([I_{i-1,j+1}, I_{i,j+1}, I_{i+1,j+1}])$
      $I'_{i,j} \leftarrow$ `median`$([col1, col2, col3])$
      $col1 \leftarrow col2$
      $col2 \leftarrow col3$
    end for
  end for
end function

	0	1	2	3	4
0	**1**	**2**	*	*	*
1	**1**	**2**	*	*	*
2	**1**	**2**	*	*	*
3	*	*	*	*	*
4	*	*	*	*	*

(a) For the first kernel of row 1, calculate the median of the first column and the second column

	0	1	2	3	4
0	**1**	**2**	**3**	*	*
1	**1**	**2**	**3**	*	*
2	**1**	**2**	**3**	*	*
3	*	*	*	*	*
4	*	*	*	*	*

(b) Calculate the median of the right-hand column and then calculate the median of the 3 column medians. This approximates the median of $I_{1,1}$

	0	1	2	3	4
0	*	**1**	**2**	**3**	*
1	*	**1**	**2**	**3**	*
2	*	**1**	**2**	**3**	*
3	*	*	*	*	*
4	*	*	*	*	*

(c) For the kernel centered at $I_{1,2}$, calculate the median of the right-hand column and then calculate the median of the 3 column medians

Table 1: An illustration of the steps in DP for a 3x3 kernel

arbitrary kernel (even ones that are not square), the proofs presented here consider only square kernels of odd side length (W is odd)[2].

DP may not find the exact median of every kernel, but it will find a value within known bounds. If L represents the sorted values of a given kernel, DP will find a value between the αth element of L and the βth element of L, where

$$\alpha = \frac{(W+1)^2}{4} - 1 \tag{1}$$

$$\beta = W^2 - \alpha = \frac{(3W-1)^2}{12} + \frac{2}{3} \tag{2}$$

3.1 Proof of accuracy

In order to prove this constraint, we define the median of an ordered (ascending) set of elements $A = \{A_0, A_1, ..., A_n\}$ as

$$median(A) = A_{\lfloor \frac{n}{2} \rfloor}. \tag{3}$$

We define the median of any set S as the median of the ordered (ascending) set containing the same values as S.

Lemma 1 follows immediately from this definition.

Lemma 1. *The median element in a distinct-valued set S of size n is greater than at least $\lfloor \frac{n}{2} \rfloor$ other members of S.*

Proposition 1 (Median of medians)**.** *If M is an indexed set such that M_i is the median value of the set m_i, with all m_i having distinct values*[3] *and equal cardinality, then the median of M is greater than* and *less than h members of $\bigcup m$, where $h = \lfloor \frac{|M|}{2} \rfloor \lfloor \frac{|m_0|}{2} \rfloor + \lfloor \frac{|m_0|}{2} \rfloor + \lfloor \frac{|M|}{2} \rfloor$.*

Proof. Let G be the set of all m_i such that $median(m_i)$ is less than $median(M)$.

$$G = \{x \mid x = m_i \wedge median(x) < median(M)\}$$

By Lemma 1, $median(M)$ is greater than at least $\lfloor \frac{|M|}{2} \rfloor$ other elements of M.

[2]In practice, W is almost always odd so that there is an exact center of the kernel [4].
[3]Note that all m_i have distinct values from each othe.r

$$|G| \geq \left\lfloor \frac{|M|}{2} \right\rfloor \qquad \text{(By Lemma 1)}$$

By Lemma 1, the median of G_i is greater than at least $\lfloor \frac{|G_i|}{2} \rfloor$ members of G_i.

$$\forall i \ \left(|\{x \mid x \in G_i \wedge x < median(G_i)\}| \geq \left\lfloor \frac{|G_i|}{2} \right\rfloor \right) \qquad \text{(By Lemma 1)}$$

Because inequalities are transitive, any value that is less than $median(G_i)$ is also less than $median(M)$.

$$\forall i \ \ (x \in G_i \wedge x < median(G_i) \to x < median(M)) \qquad \text{(By transitivity)}$$

Thus:

1. $median(M)$ is greater than at least $\lfloor \frac{|M|}{2} \rfloor$ members of M (by Lemma 1),
2. Each of those members of M are greater than at least $\lfloor \frac{|G_i|}{2} \rfloor$ members of some G_i (shown above),
3. Since $median(M)$ is itself the median of some m_i, $median(M)$ is greater than at least $\lfloor \frac{|m_i|}{2} \rfloor$ members of that m_i (by Lemma 1).

Therefore, $median(M)$ is greater than at least $\lfloor \frac{|M|}{2} \rfloor \lfloor \frac{|m_0|}{2} \rfloor + \lfloor \frac{|m_0|}{2} \rfloor + \lfloor \frac{|M|}{2} \rfloor$ members of $\bigcup m$.

$$|\{x \mid x \in \bigcup m \wedge x < median(M)\}| \geq \left\lfloor \frac{|M|}{2} \right\rfloor \left\lfloor \frac{|m_0|}{2} \right\rfloor + \left\lfloor \frac{|m_0|}{2} \right\rfloor + \left\lfloor \frac{|M|}{2} \right\rfloor$$

It follows immediately that $median(M)$ is also less than at least $\lfloor \frac{|M|}{2} \rfloor \lfloor \frac{|m_0|}{2} \rfloor + \lfloor \frac{|m_0|}{2} \rfloor + \lfloor \frac{|M|}{2} \rfloor$ members of $\bigcup m$.

$\square$

Table 2: Sample α and β values for popular median filter kernel sizes

Kernel size	α	β	Max index error
3x3	3	6	1 in 9
5x5	8	17	4 in 25
7x7	15	34	9 in 49

Algorithm 1 finds the median of W values, where each value is itself the median of W other values. Thus, proposition 1 can be applied with $|M| = |m_0| = W$. Note, however, that the proof above assumes that all values are distinct. While this is not the case in practice, the same logic still applies. Proposition 1 means that the median value selected by algorithm 1 for a pixel $I_{i,j}$ will be greater than and less than h of pixel $I_{i,j}$'s neighbors. Note that:

$$h = \left\lfloor \frac{|M|}{2} \right\rfloor \left\lfloor \frac{|m_0|}{2} \right\rfloor + \left\lfloor \frac{|m_0|}{2} \right\rfloor + \left\lfloor \frac{|M|}{2} \right\rfloor = \left\lfloor \frac{W}{2} \right\rfloor^2 + 2\left\lfloor \frac{W}{2} \right\rfloor. \tag{4}$$

And when W is odd:

$$\left\lfloor \frac{W}{2} \right\rfloor^2 + 2\left\lfloor \frac{W}{2} \right\rfloor = \left(\frac{W-1}{2}\right)^2 + 2\frac{W-1}{2} = \frac{(W+1)^2}{4} - 1 = \alpha \tag{5}$$

Since the selected value for a given pixel must be greater than at least α of its neighbors and less than at least α of its neighbors, the selected value's index in the ordered list L is between α and $W^2 - \alpha = \beta$. $\square$

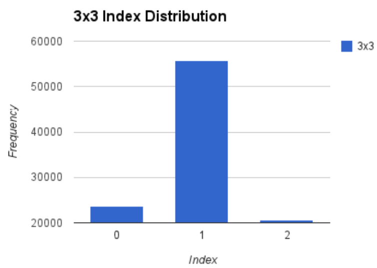
Values for popular median filter kernel sizes are shown in table 2.

3.2 Distribution

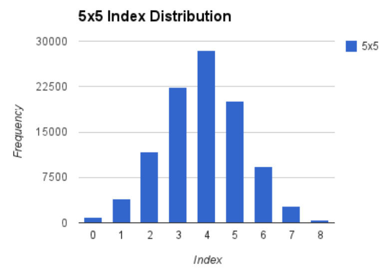
Although the distribution of the index selected as the median by DP will vary with data, testing DP with randomly generated images suggests that the value selected is not uniformly distributed between α and β. Figures 1a, 1b, and 1c show the distribution of the selected index for $n = 100000$ samples.

Figure 1d shows the cumulative error probability of the 7x7 median filter. The graph also shows a normal cumulative density function with $\mu = 1.0364$ and $\sigma = 1.4169$. In the 7x7 case (with random image data), table 3 shows that DP selects a value within one index of the median 51.9% of the time. DP selects a value within 4 indexes of median 96.2% of the time.

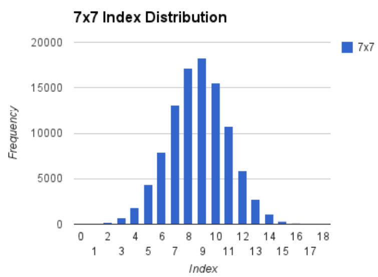
Figure 1: Distributions for various kernels



(a) 3x3 distribution



(b) 5x5 distribution



(c) 7x7 distribution

(d) 7x7 cumulative error distribution

Table 3: Cumulative error probabilities for 7x7 filter

Index error $\leq$	Probability
0	0.204
1	0.519
2	0.755
3	0.895
4	0.962
5	0.989
6	0.997
7	0.999
8	0.999
9	1.000

Table 4: Benchmark data of several median filter algorithms on an AMD Opteron 6168 (8-bit images)

Size	Bubble	Huang	Perreault and Hebert	DP	Dong	Narendra
5000^2	1.364s	5.693s	7.959s	0.46s	3.598s	0.653s
4000^2	0.868s	3.628s	4.981s	0.295s	2.317s	0.407s
3000^2	0.492s	2.043s	2.749s	0.168s	1.295s	0.231s
2000^2	0.219s	0.900s	1.189s	0.069s	0.568s	0.094s
1000^2	0.057s	0.225s	0.290s	0.018s	0.057s	0.024s

4 Performance

An implementation of the DP algorithm written in *C* was benchmarked against several other median filter implementations using a 3x3 kernel. Raw timings are shown in table 4 and graphed in figure 2.

The bubble algorithm is a hard-coded bubble-sort approach presented by Kopp and Purgathofer [7]. Huang's algorithm consists of a moving histogram kernel [5]. Perreault and Herbet present a variation of Huang's algorithm that only accesses each pixel once[9]. DP is the algorithm discussed in this paper. This implementation of Dong's algorithm [3] is slightly modified to produce results within the same bounds as DP using the sign test method. Narendra's algorithm involves a 1D median filter that can be used to approximate a 2D median filter [8].

As the bubble algorithm, Huang, and Perreault and Herbet are exact median filters that calculate the precise median of a pixel's neighborhood, comparisions to the DP, Dong and Narendra approximation algorithms are far from fair. It is also worth noting that the performance of the bubble algorithm decreases quickly and substantially as kernel size increases.

The performance of DP compares favorably to all other algorithms tested, although there are several other approximation algorithms not considered in this paper [2, 1, 12].

5 Conclusion

We have presented a new median filter algorithm that performs favorably compared to many preexisting algorithms. Unlike many of such algorithms, DP does not require data to be discretized or otherwise placed into a his-

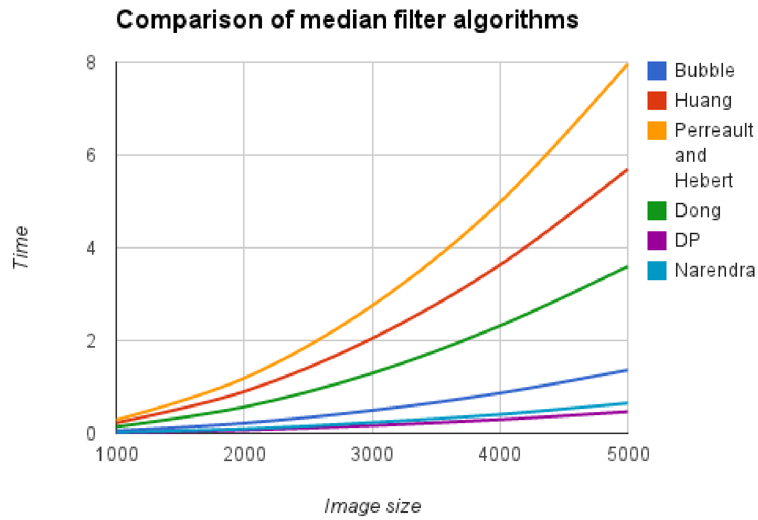


Figure 2: A graph of table 4

togram. While DP is not completely precise, the error bounds of DP are well defined, proven, and small enough for many applications.

References

[1] Kaoru Arakawa. Median filter based on fuzzy rules and its application to image restoration. *Fuzzy sets and systems*, 77(1):3–13, 1996.

[2] M. Barni, V. Cappellini, and A. Mecocci. Fast vector median filter based on euclidean norm approximation. *Signal Processing Letters, IEEE*, 1(6):92–94, 1994.

[3] Fu guo Dong. A novel fast algorithm for image median filtering based on improved sign test method. In *Logistics Systems and Intelligent Management, 2010 International Conference on*, volume 2, pages 1240–1242, 2010.

[4] Abdessamad Ben Hamza, Pedro L. Luque-Escamilla, José Martínez-Aroza, and Ramón Román-Roldán. Removing noise and preserving details with relaxed median filters. *J. Math. Imaging Vis.*, 11(2):161–177, October 1999.

[5] T. Huang, G. Yang, and G. Tang. A fast two-dimensional median filtering algorithm. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 27(1):13–18, 1979.

[6] A. C. Kak and M. Slaney. *Principles of Computerized Tomographic Imaging*. IEEE Press, New York, 1988.

[7] Manfred Kopp and Werner Purgathofer. Efficient 3x3 median filter computations. Technical report, 1994.

[8] Patrenahalli M. Narendra. A separable median filter for image noise smoothing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-3(1):20–29, 1981.

[9] S. Perreault and P. Hebert. Median filtering in constant time. *Image Processing, IEEE Transactions on*, 16(9):2389–2394, 2007.

[10] John C. Russ. *Image Processing Handbook, Fourth Edition*. CRC Press, Inc., Boca Raton, FL, USA, 4th edition, 2002.

[11] Ben Weiss. Fast median and bilateral filtering. *ACM Trans. Graph.*, 25(3):519–526, July 2006.

[12] Xiahua Yang and Peng Seng Toh. Adaptive fuzzy multilevel median filter. *Image Processing, IEEE Transactions on*, 4(5):680–682, 1995.