

## 실수형 2차원 데이터를 위한 고속 미디언 필터링 알고리즘

조태훈\*

### Fast Median Filtering Algorithms for Real-Valued 2-dimensional Data

Tai-Hoon Cho\*

School of Computer Science and Engineering, Korea University of Technology and Education, Cheonan 330-708, Korea

#### 요 약

미디언 필터링은 임펄스 형태의 잡음의 제거에 매우 효과적이어서, 많은 신호처리 응용분야에서 널리 사용되어 왔다. 하지만, 비선형성에 의한 시간 복잡도로 인하여, 미디언 필터링은 주로 작은 필터윈도우 크기를 사용하였다. 고속 미디언 필터링 알고리즘에 대한 많은 연구가 진행되었지만 대부분 영상과 같은 한정된 정수값을 갖는 입력데이터에만 적용될 수 있으며, 실수형 2차원 데이터의 고속 미디언 필터링 알고리즘에 대한 연구는 미미한 실정이다. 본 논문에서는 간단하면서도 실수형 2차원 데이터를 고속으로 미디언 필터링할 수 있는 알고리즘을 제안하고 Matlab의 2차원 미디언 필터와 힙(heap)기반의 2차원 미디언 필터와 성능을 비교하였다. 다양한 필터윈도우 크기에 대해서 제안된 알고리즘이 Matlab의 필터보다는 훨씬 빠르고, 힙기반의 필터보다는 대부분 일관되게 더 빠른 결과를 내었다. 또한, 한정된 데이터 값 범위를 갖는 실수형 2차원 데이터는 비트수가 큰 정수형 고속 2차원 미디언 필터링 알고리즘을 이용하여 거의 오차없이 매우 빠르게 미디언 필터링을 할 수 있음을 보였다.

#### ABSTRACT

Median filtering is very effective to remove impulse type noises, so it has been widely used in many signal processing applications. However, due to the time complexity of its non-linearity, median filtering is often used using a small filter window size. A lot of work has been done on devising fast median filtering algorithms, but most of them can be efficiently applied to input data with finite integer values like images. Little work has been carried out on fast 2-d median filtering algorithms that can deal with real-valued 2-d data. In this paper, a fast and simple median 2-d filter is presented, and its performance is compared with the Matlab's 2-d median filter and a heap-based 2-d median filter. The proposed algorithm is shown to be much faster than the Matlab's 2-d median filter and consistently faster than the heap-based algorithm that is much more complicated than the proposed one. Also, a more efficient median filtering scheme for 2-d real valued data with a finite range of values is presented that uses higher-bit integer 2-d median filtering with negligible quantization errors.

**키워드** : 미디언 필터링, 2차원 필터, 힙, 영상처리

**Key word** : Median filtering, 2-d filter, heap, image processing

접수일자 : 2014. 07. 29 심사완료일자 : 2014. 09. 05 게재확정일자 : 2014. 09. 22

\* **Corresponding Author** Tai-Hoon Cho (E-mail: thcho@koreatech.ac.kr, Tel: +82-41-560-1351)

School of Computer Science and Engineering, Korea University of Technology and Education, Cheonan 330-708, Korea

**Open Access** <http://dx.doi.org/10.6109/jkiice.2014.18.11.2715>

print ISSN: 2234-4772 online ISSN: 2288-4165

© This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.  
Copyright © The Korea Institute of Information and Communication Engineering.

## I. 서 론

미디언 필터링(median filtering)은 평균값 필터링과 함께 1차원 또는 영상처리 등의 2차원 신호처리에서 널리 사용되어왔다[1]. 평균값 필터링은 랜덤 노이즈(random noise)의 제거에 유용한 반면에, 미디언 필터링은 임펄스(impulse) 형태의 잡음 제거에 효과적이다. 하지만, 미디언 필터링은 기본적으로 데이터 값의 크기순으로 정렬이 요구되기 때문에 필터 윈도우(filter window) 크기가 커짐에 따라 급격히 계산 시간이 증가하므로 실제 적용시 작은 필터 윈도우 크기를 많이 사용된다.

따라서, 고속 미디언 필터링 알고리즘에 관한 많은 연구가 수행되어 왔다. 하지만, 대부분[2-4]은 8비트 또는 16비트 등의 고정된 픽셀 깊이(pixel depth)를 갖는 데이터 값이 정수인 영상에 적용될 수 있는 히스토그램(histogram) 기반 필터링 알고리즘이다. 실수값을 갖는 2차원 데이터를 다룰 수 있는 고속 미디언 필터링 알고리즘에 대한 연구는 미미한 실정이다[5].

본 논문에서는 실수형 데이터를 처리할 수 있는 간단하면서도 고속인 2차원 미디언 필터를 제안하고 다른 두 가지 방법과 실행시간을 비교하였다. 하나는 많이 사용되는 상용 툴인 Matlab의 2차원 미디언 필터이고, 다른 하나는 최소 힙(min heap)과 최대 힙(max heap)[6]을 이용한 힙 기반 1차원 미디언 필터[7]를 확장하여 구현한 힙 기반 2차원 미디언 필터이다.

비교 결과 제안된 2차원 미디언 필터가 가장 속도가 빠름을 나타내었다. 하지만, 8비트 등의 정수형 데이터의 최근의 고속 2차원 미디언 필터[3,4]와 비교해서는 속도가 현저하게 느리다.

따라서 본 논문에서는 8비트 영상에 대해서만 적용되는 [4]의 공개 소스 코드[8]를 수정하여 8, 10, 12, 14, 16비트 정수형 2차원 데이터에 적용가능한 고속 2차원 미디언 필터를 구현하였다. 이를 이용하여 실수형 2차원 데이터를 완전한 미디언 필터결과와 오차가 거의 나지 않게 하면서 미디언 필터링을 고속으로 수행할 수 있음을 보였다.

2장에서는 실수형 2차원 미디언 필터링을 위한 제안된 알고리즘 및 정수형 고속 2차원 미디언 필터링을 이용한 데이터값의 범위가 한정된 실수형 데이터의 효율적인 미디언 필터링 방안에 대한 자세한 설명을 하고, 3

장에서 다른 알고리즘과의 성능 비교한 결과를 보인다. 마지막으로 4장에 결론을 맺는다.

## II. 알고리즘

### 2.1. 실수형 2차원 데이터의 고속 미디언 필터링

제안된 고속 2차원 미디언 필터링 알고리즘의 자세한 설명은 다음과 같다. W와 H는 각각 2차원 데이터의 너비(width)와 높이(height)를 나타낸다. N은 전체 데이터 값의 개수이다. (즉,  $N=W \times H$ ) 2차원 필터 윈도우는 각 사이트의 길이가  $2r+1$ 이 된다. 이때, r은 필터반경이다. 따라서, 필터 윈도우 내의 데이터 값들의 개수  $N_{win}$ 은  $(2r+1)^2$ 이 된다.

#### Algorithm 1

입력 :  $I(x,y)$ , 2차원 실수형 데이터 ( $W \times H$ ),  
필터반경 r  
출력 :  $O(x,y)$ , 2차원 실수형 데이터 ( $W \times H$ )

**Step 1.** 입력데이터를 오름차순으로 정렬하여,  
 $S_i, X_i, Y_i, i=1, \dots, N$  을 생성한다.  
여기서,  $S_i$ 는 정렬된 데이터에서 i번째  
데이터값,  $X_i, Y_i$ 는  $S_i$ 의 x, y 좌표를  
각각 나타낸다.

**Step 2.** 모든 x, y에 대해  $n(x,y)$ 를 0으로  
초기화한다.

**Step 3.**

**for** i=1 to N **do**

$(X_i, Y_i)$ 를 중심으로 하는  $(2r+1) \times (2r+1)$   
크기의 2차원 필터 윈도우안의

각 좌표  $(x', y')$ 에 대해서

**if**  $n(x', y') \leq N_{win}/2$  **then**

//  $N_{win} = (2r+1) \times (2r+1)$

$n(x', y') := n(x', y') + 1$

$O(x', y') := S_i$

**end if**

**end for**

위 알고리즘이 수행되면, 다음과 같은 이유로  $O(x', y')$ 은

(x', y')을 중심으로 하는 윈도우 내의 데이터의 미디언 값이 되게 된다. 윈도우 내의 데이터로부터 오름차순으로 하나씩 값을 선택함에 따라 n(x',y')은 1씩 증가하게 된다. 따라서, n(x',y')이 N<sub>win</sub>/2+1이 되면 S<sub>i</sub>는 데이터의 N<sub>win</sub>/2+1 번째 데이터값, 즉 현재 윈도우 안에 있는 데이터의 미디언이 되게 된다. 위 알고리즘에서 임계값 N<sub>win</sub>/2을 0과 N<sub>win</sub>-1으로 변경하면 각각 윈도우 내의 최소값과 최대값도 쉽게 찾을 수 있다.

위 알고리즘의 특징은 데이터 정렬은 Step 1에서 전체 데이터에 대한 정렬만 한번 이루어지고, 매 필터 윈도우마다의 정렬은 필요치 않아 매우 효율적인 계산이 가능하다.

위 알고리즘에서 Step 1의 정렬알고리즘으로 임의의 고속 정렬알고리즘을 사용할 수 있다. 여기서는 ANSI C 라이브러리의 quicksort 버전인 qsort 함수[9]를 사용하였다.

## 2.2. 정수형 고속 2차원 미디언 필터링을 이용한 실수형 2차원 데이터의 미디언 필터링

값의 범위가 한정된 실수형 2차원 데이터의 미디언 필터링은 비트 수가 큰 정수형 2차원 미디언 필터링 알고리즘을 이용하여 거의 오차없이 매우 빠르게 필터링을 할 수 있다. 정수형 미디언 필터링 알고리즘으로는 현재 가장 빠른 방법 중 하나인 [4]의 공개 소스코드[8]를 수정하여 8~16비트의 다양한 비트 수의 정수형 미디언 필터링이 가능하게 구현하였다.

### Algorithm 2

입력 : f(x,y), 2차원 실수형 데이터 (WxH),  
 필터반경 r, 픽셀 깊이(pixel depth) d  
 출력 : f<sub>m</sub>(x,y), 2차원 실수형 데이터 (WxH)

**Step 1.** 실수형 입력데이터 f를 정수형 데이터 I로 양자화한다.

f<sub>min</sub> : 입력 데이터 최소값

f<sub>max</sub> : 입력 데이터 최대값

MAX\_NUM : 2<sup>d</sup>-1

NINT(.) : 반올림 연산자

$$I(x,y) = \text{NINT}\left(\frac{f(x,y) - f_{\min}}{f_{\max} - f_{\min}} \cdot \text{MAX\_NUM}\right)$$

**Step 2.** I(x,y)를 d-비트 정수형 미디언필터링을 하여 I<sub>m</sub>(x,y)를 생성한다.

**Step 3.** I<sub>m</sub>을 다시 실수형 출력데이터 f<sub>m</sub>으로 변환한다.

$$f_m(x,y) = f_{\min} + I_m(x,y) \frac{f_{\max} - f_{\min}}{\text{MAX\_NUM}}$$

Step 1에서 양자화 해상도(quantization resolution) Q는

$$Q = \frac{f_{\max} - f_{\min}}{\text{MAX\_NUM}}$$

로 주어지며, 양자화시 최대 오차는 Q/2가 된다.

## III. 실험 결과

먼저, 실수형 2차원 데이터의 고속 미디언 필터링을 위한 제안된 알고리즘(Algorithm 1)의 성능을 평가하기 위해 두 가지 방법과 비교하였다. 하나는 Matlab 2014a 버전에서 제공되는 2차원 미디언필터 *medfilt2*이고, 다른 하나는 최소 힙(min heap)과 최대 힙(max heap)[6]을 이용한 힙 기반 1차원 미디언 필터[7]를 확장하여 구현한 힙 기반 2차원 미디언 필터이다.

제안된 알고리즘과 힙기반 알고리즘은 윈도우즈 7 운영체제의 애플 iMAC에서 MS Visual C++ 2008으로 구현하여, 테스트하였다. 사용된 입력데이터는 실수값을 갖는 너비x높이가 640x480인 위상차 (phase difference) 데이터이다. 이 데이터는 위상측정법에 의한 3차원 측정시스템에서 얻어진 위상차 데이터이다. 그림 1은 입력데이터를 보여준다. 입력데이터 값의 범위는 -1.517192~9.153144이고, 각 값은 소숫점 6자리까지 주어지는 float 값으로 되어있다. 그림 2는 필터반경 r=15을 갖는 31x31의 필터윈도우를 사용하여 미디언 필터링한 결과를 보여주고 있다.

그림 3은 제안된 알고리즘, 힙기반 알고리즘, 그리고 Matlab 2차원 필터의 실행시간을 필터반경이 커짐에 따라 보여준다. 제안된 알고리즘과 힙기반 알고리즘이 Matlab 2차원 필터에 비해 상당히 큰 필터윈도우에 대해서 매우 빠른 것을 알 수 있다. 전반적으로 제안된 알고리즘이 힙기반 알고리즘보다 빠른 결과를 보여준다.

제안된 알고리즘의 실행 시간 대비 다른 알고리즘의 실행시간 비율을 표 1에 표시하였다. Matlab 2차원 필터는 약 3~5배 정도 더 걸리고, 힙 기반 알고리즘은 r이

커짐에 따라 제안된 알고리즘 보다 약 30~70% 정도 오래 걸리는 것을 알 수 있다.

참고로  $r=15$ 일 때 제안된 알고리즘은 약 0.38초 걸리는데 비해, 매 필터링 윈도우에서 ANSI C 라이브러리 함수인 qsort로 정렬하여 미디언값을 구하면 약 24.7초로 매우 오래 걸린다.

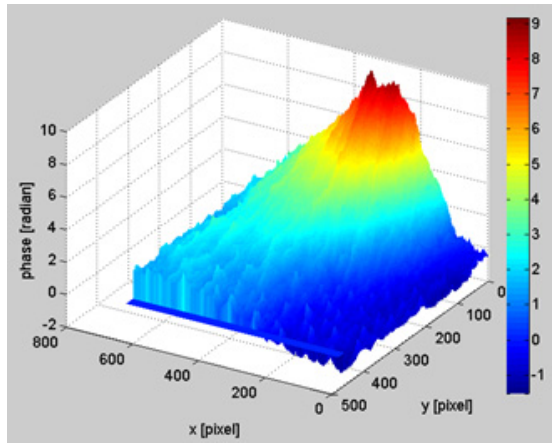


그림 1. 실수값을 갖는 2차원 위상데이터 입력  
Fig. 1 Input 2-d phase data whose values are real-valued

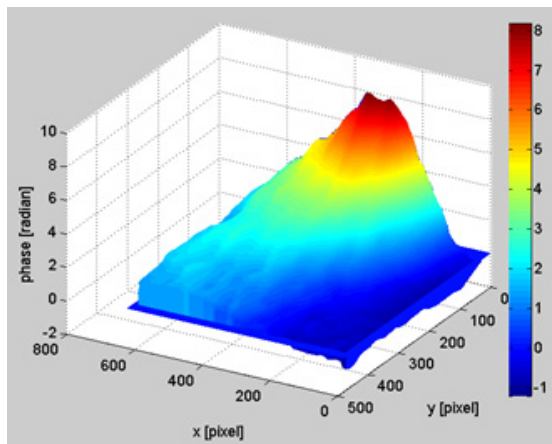


그림 2. 2차원 미디언 필터링 출력  
Fig. 2 Output of 2-d median filtering

8, 10, 12, 14, 16비트 정수형 2차원 데이터에 적용가능하도록 수정된 알고리즘을 이용하여 그림 1의 위상차 데이터를 미디언 필터링( $r=15$ )하여 얻은 결과의 오차와 실행 시간을 표 2에 나타내었다. 오차는 실수형 데이터

2차원 미디언 필터를 적용한 정확한 미디언 필터링 출력과의 차이를 말한다. 비트수가 커질수록 양자화오차가 작아져서, 최대 오차, RMSE(Root Mean Squared Error)가 줄어들지만, 실행 시간은 증가하는 것을 볼 수 있다. 이론적으로 추정되는 바와 같이 2 비트 증가하면 최대오차는 1/4로 감소하는 것을 알 수 있다.

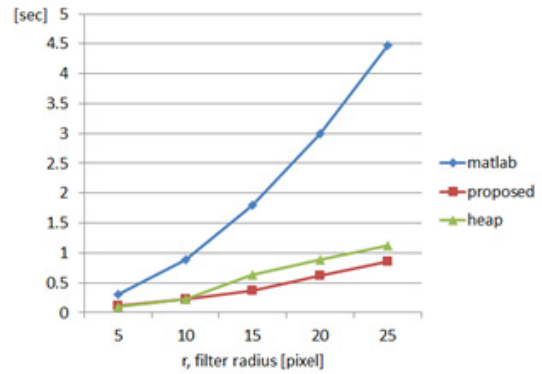


그림 3. 다른 알고리즘과의 비교한 제안된 알고리즘의 미디언 필터링 시간  
Fig. 3 Median filtering time of the proposed method compared with other algorithms

표 1. 필터링 수행 시간 비 (제안 알고리즘 시간 기준)  
Table. 1 Filtering time ratio (reference : proposed algorithm)

r	제안된 알고리즘	Matlab	Heap
5	1.0	2.82	0.91
10	1.0	4.00	1.05
15	1.0	4.86	1.73
20	1.0	4.84	1.42
25	1.0	5.27	1.32

8비트에서 10비트로 커지면 RMSE는 0.02에서 0.003정도로 1/7정도로 감소하는 반면, 실행시간은 0.023초에서 0.04초로 약 1.7배 늘어나는 것을 알 수 있다. 10비트에서 12비트로 커지면 RMSE는 1/3정도로 감소하고, 실행시간은 2배 정도 늘어난다. 14, 16비트가 되면 오차는 더욱 감소하지만, 실행시간이 각각 0.3, 0.5초로 급격히 증가하는 것을 볼 수 있다. 따라서, 10 또는 12비트 정수형 미디언 필터링 알고리즘을 사용하는 것이 실질적으로 오차를 최소화하면서 속도를 극대화할 수 있다. 10비트 인 경우 실행시간이 0.04초로 앞서 제

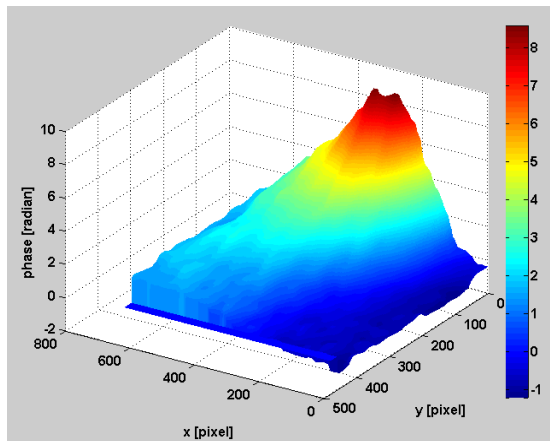
안된 실수형 데이터 2차원 미디언 필터링 시간 0.38초보다 거의 10배 정도 빠른 것을 알 수 있다.

참고로, 그림 4에 10비트 정수형 미디언 필터( $r=15$ )를 그림 1의 위상차 데이터에 적용하여 얻은 결과를 보였다. 그림 2의 정확한 2차원 미디언 필터링 출력과 차이가 없음을 눈으로 확인할 수 있다.

**표 2.** 정수형 미디언 필터를 그림 1의 실수형 데이터에 적용 시의 오차 및 실행 시간

**Table. 2** Errors obtained by applying the integer type median filter to real-valued data of Fig. 1.

	픽셀 깊이 (pixel depth) [bits]				
	8	10	12	14	16
최대 오차	0.020941	0.005220	0.001304	0.000326	0.000082
RMSE	0.019373	0.002914	0.001185	0.000254	0.000021
실행시간 [초]	0.023	0.04	0.083	0.3	0.5



**그림 4.** 10비트 정수형 미디언 필터( $r=15$ )를 적용한 결과  
**Fig. 4** Results obtained by applying 10-bit 2-d median filter with  $r=15$

#### IV. 결론

본 논문에서는 실수값을 갖는 데이터 입력에 대한 간단하면서도 고속인 미디언 2차원필터를 제안하고 성능을 Matlab 2차원 필터와 힙기반의 2차원 미디언 필터와 비교하였다. 제안된 알고리즘은 Matlab 2차원 필터보

다 훨씬 빠름을 보였고, 제안된 알고리즘보다 훨씬 복잡한 힙기반의 알고리즘보다는 일관되게 빠른 결과를 보였다. 또한, 실수형 입력데이터의 범위가 한정된다면 8, 10, 12 비트 정수형 데이터 미디언 필터를 이용하여 원하는 허용오차 범위 내에서 실수형 2차원 데이터의 미디언 필터링을 더욱 빠르게 실행할 수 있음을 보였다.

#### 감사의 글

이 논문은 2013년도 한국기술교육대학교 연구재단 과제연구비 지원에 의하여 연구되었음

#### REFERENCES

- [1] R. Crane, *A Simplified Approach to Image Processing*, Prentice Hall, 1996.
- [2] T. Huang, G. Yang, and G. Tang, "A Fast two-dimensional median filtering algorithm," *IEEE Trans. on Acoust., Speech, Signal Process.*, vol. 27, no. 1, pp. 13-18, 1979.
- [3] B. Weiss, "Fast median and bilateral filtering," *ACM Trans. on Graphics (TOG)*, vol. 25, no. 3, pp. 519-526, 2006.
- [4] S. Perreault, "Median filtering in constant time," *IEEE Trans. on Image Processing*, vol. 16, no. 9, pp. 2389-2394, 2007.
- [5] J. Gil and M. Werman, "Computing 2-d min, median, and max filters," *IEEE Trans. on Pattern Anal. Machine Intell.*, vol. 15, no. 5, pp. 504-507, 1993.
- [6] M. Juhola, J. Katajainen, and T. Raita, "Comparison of algorithms for standard median filtering," *IEEE Trans. on Signal Processing*, vol. 39, no. 1, pp. 204-208, 1991.
- [7] S.L. Jensen, "Implantable medical device fast median filter," *US Patent 6,941,332 B2*, 2005.
- [8] <http://nomis80.org/ctmf.html>.
- [9] B.W. Kernighan and D.M. Ritchie, *The C programming language*, 2nd. ed. Prentice Hall, 1988.



**조태훈 (Tai-Hoon Cho)**

1991. 5 : Virginia Tech 전기공학과 공학박사  
1992. 3 ~ 1998. 2 : LG산전연구소 수석연구원  
1998. 3 ~ 현재 : 한국기술교육대학교 컴퓨터공학부 교수  
※ 관심분야 : 영상처리, 컴퓨터비전