

# 효율적인 하드웨어 구현을 위한 정렬 알고리즘에 대한 분석

## Analysis of Sorting Algorithm for Efficient Hardware Implementation

김 한 결\*, 강 봉 순\*<sup>★</sup>

Han Kyeol Kim\*, Bongsoon Kang\*<sup>★</sup>

### Abstract

Under the influence of Autonomous Driving and AI, it is important to accurately recognize and judge objects through cameras. In particular, since a method of recognizing an object using a camera can obtain a large amount of information visually compared to other methods, many image signal processing methods have been studied to extract an accurate image. In addition, a lot of research is being carried out to implementation about hardware. In this work, we compare the principles and characteristics of the sorting algorithms that are frequently used in image signal processing and summarize the performance evaluation. Based on this, we define an efficient algorithm when implemented in hardware among efficient sorting algorithms.

### 요 약

자율주행, AI의 시대가 도래함에 따라 카메라를 통하여 물체를 정확히 인식 및 판단하는 것이 중요해졌다. 특히 카메라를 이용하여 물체를 인식하는 방법은 다른 여러 방법들에 비하여 시각적으로 많은 양의 정보를 얻을 수 있기 때문에 정확한 영상을 추출하기 위하여 많은 영상 신호 처리 방법들이 연구되고 있다. 또한, 이러한 영상 신호 처리의 기능을 실제 하드웨어로 구현하기 위하여 많은 연구도 동시에 진행되고 있다. 본 논문에서는 영상 신호 처리에서 자주 사용되는 정렬 알고리즘에 대하여 동작원리 및 특징을 비교하고 성능에 대한 평가를 정리하였다. 이를 토대로 대표적인 정렬 알고리즘 중 하드웨어로 구현할 때 효율적인 알고리즘에 대하여 정의한다.

*Key words : Quick Sort, Batch Sort, Big-O-notation, Hardware Implementation, Area*

### 1. 서론

최근 자율주행(Autonomous Driving)에 대한 관심이 높아지면서 많은 연구기관에서 자율주행에 관한 기술개발 및 상용화를 위한 연구를 진행하고 있다. 자율주행에서 객체를 인식하는 방법에는 크

게 LiDAR(Light Detection And Range)와 카메라를 이용한 컴퓨터 비전기술이 존재한다. LiDAR 방법은 빛을 이용하여 주변 물체와의 거리를 측정하는 방법이고, 컴퓨터 비전기술은 차량에 장착되어 있는 카메라를 통하여 차선 및 주변 물체에 대한 정보를 획득하게 된다[1-2]. 컴퓨터 비전기술을 이

\* Dept. of Electronic Engineering, Dong-A University

★ Corresponding author

E-mail : bongsoon@dau.ac.kr, Tel : +82-51-200-7703

※ Acknowledgment

This paper was supported by research funds from Dong-A University.

Manuscript received Sep. 5, 2019; revised Sep. 25, 2019; accepted Sep. 26, 2019.

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.

용하여 정보를 획득하는 방법의 장점은 카메라를 통해 다른 센서 보다 많은 정보를 획득할 수 있지만, 미세먼지, 안개, 해무, 황사와 같은 급격한 날씨 변화가 발생하는 경우 정확한 정보를 취득할 수 없다는 것이 단점이다[3]. 이를 해결하기 위하여 디지털 영상신호처리를 이용하여 급격한 날씨변화에 의한 열화현상을 극복하고자 많은 연구들이 진행되고 있다[4-5]. 많은 영상처리 기법 중에서 필터(Filter)를 사용하여 영상의 화질을 개선하고자 한다[6]. 특히 상황에 따라서 필터를 사용할 때 내부 데이터들을 정렬해야할 필요성이 있다. 정렬 알고리즘은 정렬하는 방식과 목적에 따라 많은 정렬 알고리즘이 존재하는데 대표적으로 가장 많이 사용되는 정렬 알고리즘에는 퀵 정렬(Quick Sort)과 배처 정렬(Batcher Sort)이 있다[7].

본 논문에서는 퀵 정렬과 배처 정렬 알고리즘의 각각에 대한 동작원리에 대하여 살펴보고, 두 정렬 알고리즘의 성능평가를 위한 비교 실험결과와 결론 순으로 구성된다.

## II. 본론

### 1. 퀵 정렬 알고리즘의 동작원리

퀵 정렬 알고리즘은 기준점이 되는 pivot을 정하는 위치에 따라 여러 가지 방법이 존재한다. 본 논문에서는 pivot의 위치를 정하는 방법 중 좌측값(A[Left])을 pivot으로 정하는 방법[8]을 QS1로 명명하고 좌측 값(A[Left])와 우측 값(A[Right])의 평균값을 pivot으로 정하는 방법[9]을 QS2로 명명하여 이 두 가지 방법 각각에 대하여 설명한다.

가. 좌측 값(A[Left])을 pivot으로 하는 퀵 정렬

그림 1은 좌측 값(A[Left])을 pivot으로 하는 퀵 정렬 방법 QS1의 동작원리에 대한 순서도이다. A[N]은 N길이만큼의 숫자가 랜덤하게 배열되어 있는 배열집합을 의미한다. N1은 퀵 정렬이 수행되고 하나의 배열집합이 두 개의 배열집합으로 나누어지는데 그 중에서 앞쪽 배열집합의 길이를 의미한다. N2는 두 개의 배열집합 중 나머지 하나의 배열 길이를 의미한다. Left는 배열에서 첫 번째 요소 위치를 의미하고, Right는 배열에서 마지막 요소의 위치를 의미한다. i는 pivot을 제외한 좌측 값을 의미하고, k는 우측 값을 의미한다. SWAP(x, y)는 x와

y의 값을 비교하고, 교환하는 것을 의미한다. 퀵 정렬 방법은 pivot을 기준으로 A[i]와 A[k]가 각각 큰 값이거나 작거나 같은 값을 찾아내어 교환한다. 이후 i와 k가 같거나 i가 클 때에는 pivot과 자리 교환이 발생하게 된다. 이 때, k의 값은 고정(hold)시킨다. k를 고정하고 다시 나누어진 그룹에 대하여 좌측 값을 pivot으로 설정하고 같은 방법으로 배열(A[N])의 데이터 크기가 1이 되는 경우까지 정렬을 반복한다.

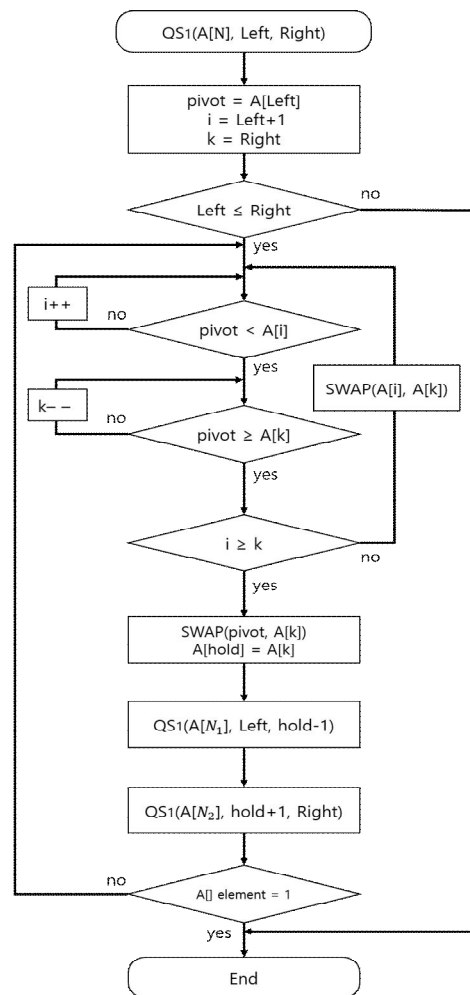


Fig. 1. Flow Chart of QS1.

그림 1.QS1의 순서도

나. 평균값을 pivot으로 하는 퀵 정렬

그림 2는 좌측 값(A[Left])과 우측 값(A[Right])의 평균값을 pivot으로 하는 퀵 정렬의 방법 QS2의 동작원리에 대한 순서도이다. A[N], N1, N2, Left, Right, SWAP(x,y)는 QS1에서의 의미와 동일하고 i는 좌측 값(A[left]), k는 우측 값(A[Right])를 의미한다. SWAP은 QS1과 동일하게 pivot을 기준

으로  $A[i]$ 와  $A[k]$ 가 각각 큰 값이거나 작거나 같은 값을 찾아내어 교환한다.

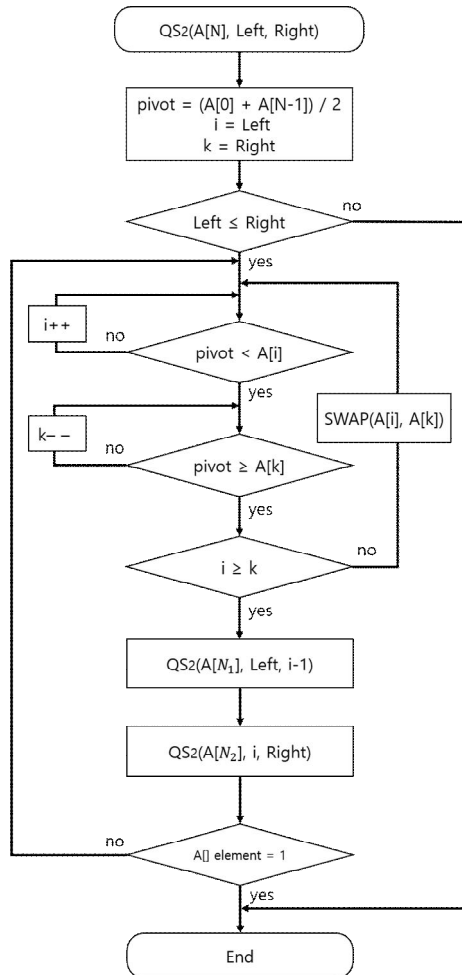


Fig. 2. Flow Chart of QS2.  
그림 2. QS2의 순서도

$i$ 가 증가하고,  $k$ 가 감소하여  $i$ 가  $k$ 보다 같거나 큰 경우에는 그 위치를 기준으로  $N_1$ ,  $N_2$  그룹으로 나눈다. 그리고 나누어진 두 그룹에 대하여 QS2의 방법을 계속 재귀적으로 사용하여 배열( $A[N]$ )의 데이터 크기가 1이 되는 경우까지 정렬한다.

**2. 배처 정렬 알고리즘의 동작원리**

배처 정렬의 방법은 입력되는 데이터의 길이  $N$ 에 의해 정렬이 이루어진다. 따라서 입력되는 데이터 순서, 또는 기준점에 관계없이 항상 일정한 비교 순서를 가진다.

그림 3은 배처 정렬 방법의 동작원리에 대한 순서도이다.  $B[N]$ 은  $N$ 길이만큼의 숫자가 랜덤하게 배열되어 있는 배열집합을 의미한다. 길이  $N$ 을 통

하여 변수  $t, p, d, r, q$ 가 정해진다.

$$i \ \& \ p = r \tag{1}$$

수식 1은 배처 정렬에서 조건식으로 사용되는 식이다. 이 조건의 의미는 변수  $i$ 와  $p$ 를 2진수로 변환하여 각 자리수마다 비트 단위로 AND 연산을 수행하는 'Bitwise AND'를 의미한다.

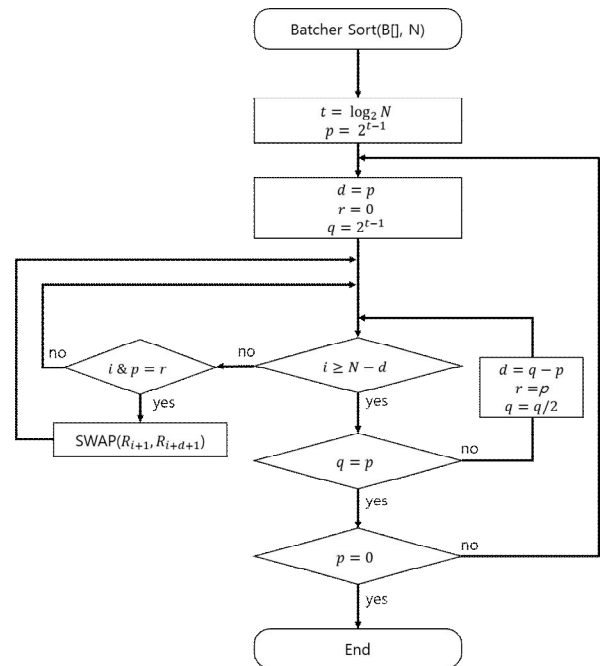


Fig. 3. Flow Chart of Batcher Sort.  
그림 3. 배처 정렬 방법의 순서도

**3. 정렬 알고리즘 대한 비교**

본 논문에서는 기준 값에 따른 퀵 정렬 QS1, QS2와 배처 정렬에 대하여 입력되는 데이터가 배열되는 순서에 따라 오름차순 또는 내림차순으로 정렬되는 SWAP(비교 및 교환)의 횟수를 비교한다. 비교 횟수는 입력되는 데이터마다 다양해질 수 있는데, 경우에 따라, QS1, QS2는 Best(최선), Worst(최악)으로 나누어질 수 있다. 배처 정렬은 오로지 데이터 길이에 따른 일정한 비교 순서를 정하는 규칙을 가지고 있다. 따라서, 배처 정렬은 Best, Worst의 경우가 존재하지 않는다. SWAP의 비교 횟수는 표 2에서 알고리즘의 성능을 평가하는 빅오 표기법(Big-O Notation)[10]을 구하는데 활용된다.

표 1은 QS1, QS2와 배처 정렬에 대하여 입력되는 데이터가 불규칙적으로 구성되어 있는 배열을 오름차순으로 정렬 할 때 소요되는 비교 횟수를

Best, Worst 경우에 대하여 기록한 표이다. 표 1에서 Data element는 입력되는 데이터의 배열 길이를 의미한다.

Table 1. Comparisons for QS1, QS2, Batcher Sort.

표 1. QS1, QS2, Batcher Sort의 비교 횟수

Data element	QS <sub>1</sub> Best	QS <sub>1</sub> Worst	QS <sub>2</sub> Best	QS <sub>2</sub> Worst	Batcher Sort
3	5	6	5	12	3
5	11	15	11	27	9
7	17	28	17	46	16
9	25	45	25	69	26
11	33	66	33	96	37
13	41	91	41	127	48
15	49	120	49	162	59
17	59	153	59	201	74
19	69	190	69	244	91
21	79	231	79	291	107
23	89	276	89	342	122
25	99	325	99	397	138
27	109	378	109	456	155
29	119	435	119	519	171
31	129	496	129	586	186

표 1에서 소요되는 횟수를 비교하면 Data element가 9이상에서는 QS1과 QS2는 Best의 경우에서는 횟수가 동일하며 가장 빠르다. 배처 정렬은 Data element가 3이상 7이하까지는 가장 빠르지만, 9이후로는 느리다. 반면에, 소요되는 횟수가 가장 느린 것은 QS2의 Worst, QS1의 Worst, 배처 정렬 순으로 구성된다.

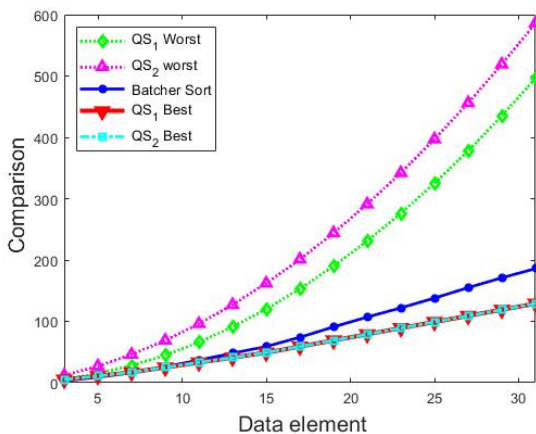


Fig. 4. Graph comparisons for QS1, QS2, Batcher sort. 그림 4. 정렬 알고리즘의 비교 횟수 그래프

그림 4는 표 1를 바탕으로 Data element가 3에서 31까지 각 정렬 방법에 대하여 비교 횟수를 나타낸 그래프이다. 그래프에서 배처 정렬은 QS1, QS2의 Worst과 Best의 사이에 위치되어 있는 것을 확인할 수 있다.

표 2는 알고리즘의 성능을 평가하는 빅오 표기법을 사용하여 퀵 정렬과 배처 정렬에 대하여 나타낸 표이다.

Table 2. Performance comparison with Big-O notation.

표 2. 빅오 표기법을 이용한 성능 비교

Method	Quick Sort(QS <sub>1</sub> , QS <sub>2</sub> )		Batcher Sort
	Best	Worst	
Big-O	$O(N \log_2 N)$	$O(N^2)$	$O(N(\log_2 N)^2)$

빅오 표기법은 알고리즘의 효율성을 평가하는 지표이다. N은 알고리즘을 수식으로 표현하였을 때, 가장 영향력이 큰 변수를 나타낸다. 따라서, N이 높을수록 시간이 많이 소요된다는 의미이다. 표 2에서 성능이 가장 좋은 알고리즘은 퀵 정렬에서 Best의 경우이지만, 성능이 안 좋은 알고리즘은 퀵 정렬의 Worst이다.

Table 3. Comparisons using Median Filter.

표 3. 중간 값 필터 사용 시 비교 횟수 (단위: 십만)

Size	QS <sub>1</sub> Best	QS <sub>1</sub> Worst	QS <sub>2</sub> Best	QS <sub>2</sub> Worst	Batcher Sort
320x240	7.6	25.0	7.6	30.5	10.6
640x480	30.4	99.8	30.4	122.0	42.4
800x600	47.5	156.0	47.5	190.6	66.2
1024x768	77.9	255.6	77.9	312.2	108.5
1920x1080	205.3	673.9	205.3	823.2	286.2
2560x1440	365.0	1,198.0	365.0	1,463.5	508.7
3840x1600	608.3	1,996.8	608.3	2,439.2	847.9
3840x2160	821.1	2,695.7	821.1	3,292.9	1,144.6

표 3와 그림 5는 단일 영상에서 자주 사용되는 영상 크기에 데이터의 중간 값을 사용하는 Median Filter를 적용하고 각 정렬 알고리즘에 대한 비교 횟수를 나타낸 것이다. Median Filter의 윈도우(Window) 크기는 5-by-5로 실험하였다. 실험 결과 Best의 경우에는 퀵 정렬의 속도가 빠르지만, Worst의 경우에는 배처 정렬보다 느린 것을 확인할 수 있었다. 단일 영상이 아닌 동영상의 경우는

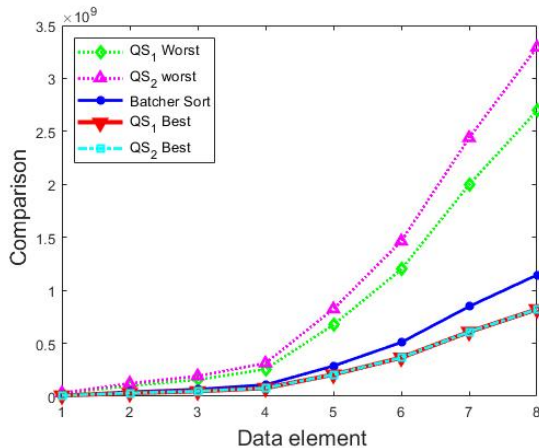


Fig. 5. Graph comparisons using Median Filter.

그림 5. 중간 값 필터 사용 시 비교 횟수 그래프

Best, Worst의 경우가 혼합인 경우가 일반적일 것이다. 이러한 경우 오히려 퀵 정렬을 사용하는 경우가 더 느리게 동작할 수 있다.

### III. 결론

본 논문에서는 정렬 알고리즘에서 소프트웨어와 하드웨어에서 자주 사용되는 퀵 정렬 알고리즘 중 QS<sub>1</sub>과 QS<sub>2</sub>, 그리고 배처 정렬에 대하여 정렬되는 과정에서 발생하는 비교 횟수를 측정하고 Median Filter에 적용시켜보았다. Best의 경우에는 퀵 정렬 알고리즘의 경우가 빠르지만, Worst의 경우에는 퀵 정렬 알고리즘이 배처 정렬보다 느린 경우를 확인할 수 있다. 더 나아가 동영상의 경우 입력되는 데이터가 다양하기 때문에 항상 Best의 경우가 나오는 경우는 거의 불가능하다. 따라서, 입력되는 데이터에는 상관없이 데이터의 길이에만 의존하는 배처 정렬의 경우가 동영상과 같이 많은 양의 데이터가 입력될 때, 속도측면에서 더 나은 성능을 기대할 수 있다. 또한, 정렬 알고리즘을 하드웨어로 구현하는 경우, 하드웨어는 모든 경우의 수를 고려하여 설계사양을 정해야 한다. 이로 인해 하드웨어의 크기는 Worst의 경우로 설계해야 한다. 다시 말해, QS<sub>1</sub>, QS<sub>2</sub>와 같이 정렬 알고리즘 중에서 속도가 빠르다고 알려진 퀵 정렬 알고리즘을 하드웨어로 구현하면 가능한 경우의 수를 최악의 조건으로 고려해야 하기 때문에 하드웨어의 크기가 커지게 된다. 반면에 배처 정렬의 경우는 입력되는 데이터 배열순서와 상관없이 길이에만 의존하기 때문에 항상 동일한 비교 횟수를 가진다. 따라서 배처 정

렬을 하드웨어로 구현하게 되는 경우 퀵 정렬의 Worst의 경우보다 더 적은 크기로 하드웨어를 구성할 수 있다.

### References

- [1] T. Wu and A. Ranganathan, "Vehicle localization using road markings," *2013 IEEE Intelligent Vehicles Symposium(IV)*, pp.1185-1190, 2013. DOI: 10.1109/IVS.2013.6629627
- [2] S. J. Han and J. D. Choi, "Real-Time Precision Vehicle Localization Using Numerical Maps," *ETRI Journal*, Vol.36, no.6, pp.968-978, 2014. DOI: 10.4218/etrij.14.0114.0040
- [3] C. Park, Seok-Cheol Kee, "Implementation of Autonomous Driving System in the Intersection Area Equipped with Traffic Lights," *Transactions of Korean Society of Automotive Engineers*, Vol.27, no.5, pp.379-387, 2019. DOI: <https://doi.org/10.7467/KSAE.2019.27.5.379>
- [4] Geun-Jun Kim, Seungmin Lee, and Bongsoon Kang, "Single Image Haze Removal Using Hazy Particle Maps," *IEICE Trans. Fundamental*, vol.E101-A, no.11, pp.1999-2002, 2018. DOI: 10.1587/transfun.E101.A.1999
- [5] Q. Zhu, J. Mai and L. Shao, "A Fast Single Image Haze Removal Algorithm Using Color Attenuation Prior," *IEEE Transaction on Image Processing*, vol.24, no.11, pp.3522-3533, 2015. DOI: 10.1109/TIP.2015.2446191
- [6] Chan-Geun Park and Byung In Choi, "The effective noise reduction method in infrared image using bilateral filter based on median value," *Journal of the Korea Society of Computer and Information*, vol.21, no.12, pp.27-33, 2016. DOI: 10.9708/JKSCI.2016.21.12.027
- [7] Donald E. Knuth, "*The Art of Computer Programming : Volume3 : Sorting and Searching*," Addison-Wesley, 1998.
- [8] S. Park and B. Kang, "Comparison of Sorting Algorithms for Improving Image Processing Speed," *The Korean Institute of Communications and Information Sciences*, pp.51-52, 2019.

[9] J. Lee and B. Kang, "Comparison of Sorting Algorithms for Improving Filter Performance," *Korea Multimedia Society(KMMS)*, pp.151-152, 2019.

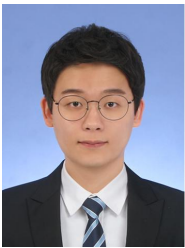
[10] D. E. Knuth, "Big Omicron and Big Omega and Big Theta," *SIGACT News*, pp.18-24, 1976.

---

## BIOGRAPHY

---

**Han Kyeol Kim** (Member)



2018 : BS degree in Electronic Engineering, Dong-A University.  
2018~ : MS degree course in Electronic Engineering, Dong-A University.

**Bongsoon Kang** (Member)



1985 : BS degree in Electronic Engineering, Yonsei University.  
1987 : MS degree in Electrical Engineering, University of Pennsylvania.  
1990 : PhD degree in Electrical Engineering, Drexel University.

1989~1999 : Senior Staff Researcher, Samsung Electronics.

1999~ : Prof. of Dept. of Electronic Engineering, Dong-A University.