

GPU를 이용한 고속 스펙클 제거 알고리즘에 관한 연구

A Study on the Acceleration of a Speckle Reduction Algorithm using GPUs

이건 · 이명준* · 정남훈 · 최재호 · 김경태

Geon Lee · Myung-Jun Lee* · Nam-Hoon Jeong · Jae-Ho Choi · Kyung-Tae Kim

요 약

합성 개구면 레이더(synthetic aperture radar: SAR) 영상 생성 시, 영상의 한 해상도 셀 내에 다수의 산란자가 존재할 경우, 산란자들로부터 반사되어 돌아오는 신호가 상호간에 간섭을 일으켜 영상 내의 화소의 강도를 불규칙적으로 변화시키며, 이러한 불규칙적인 강도를 지닌 화소들을 스펙클이라고 한다. 스펙클은 화소 탐지 시 오탐지율을 증가시키고, 탐지율을 떨어트리며, 이러한 문제를 해결하기 위해 스펙클 제거 필터들이 개발되었다. 그러나 스펙클 제거 성능이 우수한 필터들은 처리 시간이 길어 실시간 처리와 같은 고속 처리가 중요한 상황에서 사용할 수 없었다. 본 논문에서는 스펙클 제거 필터의 처리 속도를 증가시키기 위하여 그래픽 처리 장치(graphics processing unit: GPU)를 사용하였으며, 기존의 스펙클 제거 필터의 알고리즘을 GPU의 병렬화 연산에 맞춰 최적화시키고, SAR 영상에 대하여 최적화한 알고리즘과 기존의 알고리즘을 수행하여 수행시간 및 필요로 하는 데이터 공간을 분석하였다.

Abstract

When a synthetic aperture radar (SAR) image is created, if there are some scatterers in the resolution cell, the reflected signals from the scatterers interfere with each other, causing irregular changes in the intensity of pixels in the image called speckles. Speckles increase the false alarm rate and decrease the pixel detection rate. Therefore, speckle reduction filters have been developed to reduce the influence of speckles. However, filters that reduce large amounts of speckles have a long processing time; therefore, they are costly to use in real-time applications, in which quick processing is crucial. This study used a graphics processing unit (GPU) to increase the processing speed of speckle reduction filters and optimized the existing speckle reduction algorithm to run in parallel with a GPU. Furthermore, we compared the processing time and data capacity of existing and optimized algorithms.

Key words: Lee Filter, Enhanced Lee Filter, Non-Local Means Filter

I. 서 론

레이다 영상은 낮과 밤에 관계없이 표적에 대한 정보를 획득할 수 있다는 장점이 존재하여 군사 및 민간용으

로 많이 사용되고 있다. 레이다 영상은 표적에 레이다 신호를 송신했을 때, 표적의 산란자(scatterer)로부터 반사되어 돌아오는 신호들을 합성하여 이차원 영상을 생성한다. 이때, 영상의 한 해상도 셀(resolution cell) 내에 다수의 산

「이 연구는 2021년도 국방과학연구소가 지원하는 연구비의 지원으로 연구되었음.」

포항공과대학교 전자전기공학부(Department of Electronic Engineering, Pohang University of Science and Technology)

*한국항공우주연구원 국가위성정보 활용지원센터(National Satellite Operation & Application Center, Korea Aerospace Research Institute)

· Manuscript received August 02, 2021 ; Revised August 16, 2021 ; Accepted August 23, 2021. (ID No. 20210802-078)

· Corresponding Author: Kyung-Tae Kim (e-mail: kkt@postech.ac.kr)

란자가 존재하게 된다면, 이들로부터 반사되어 돌아오는 신호들이 상호 간에 간섭을 일으켜 영상의 화소의 강도(intensity)가 임의의 크기만큼 변하게 되는데, 이렇게 생성되는 불규칙적인 강도를 지닌 화소들을 스펙클(speckle)이라고 한다^[1]. 생성된 스펙클은 합성 개구면 레이더(synthetic aperture radar: SAR) 영상에서 탐지 알고리즘을 수행할 시 영상의 화소 탐지율(detection) 대 오탐지율(false alarm rate) 성능을 떨어트리는 문제를 발생시킨다^[2]. 이러한 문제를 해결하기 위하여 다양한 스펙클 제거 필터들이 개발되어 왔다.

스펙클 제거 필터들은 참조하는 화소들의 위치에 따라 지역 필터(local filter)와 비지역 필터(non-local filter)로 나눌 수 있다. 지역 필터는 스펙클 제거를 위해 이웃한 화소들을 참조하여 표적 화소의 값을 조정하며, 비지역 필터는 검색 창(search window)을 설정하여 검색 창 내의 모든 화소를 참조하여 표적 화소의 값을 조정한다. 일반적으로 이웃한 화소들만을 참조하여 필터를 수행하는 지역 필터보다 검색 창 내부의 모든 화소를 참조하는 비지역 필터의 스펙클 제거 성능이 우수하지만, 참조하는 화소의 수가 많아 처리 시간이 오래 걸린다는 단점이 존재한다. 대표적인 지역 필터로는 평균(mean) 필터, 중앙값(median) 필터, 리(Lee) 필터^[3], 향상된 리(enhanced Lee) 필터^[4] 등이 있으며, 대표적인 비지역 필터에는 비지역 평균(non-local means; NL-means) 필터^[5]가 있다.

한편, 알고리즘의 수행 속도를 증가시키기 위하여 기존의 중앙처리장치(central processing unit: CPU)가 아닌 GPU를 적용한 알고리즘 연구가 활발하게 연구되고 있다. CPU는 제어(control) 및 캐시(cache)에 할당된 공간이 많으며, 산술 논리 장치(arithmetic logic unit: ALU)의 연산 속도가 우수한 대신 적은 수의 ALU를 가지고 있어 복잡한 구조의 연산을 수행하는 데에 적합하며, GPU는 제어 및 캐시에 할당된 용량이 적고 ALU 하나의 연산 속도가 떨어지는 대신 많은 수의 ALU가 있어 단순한 형태의 연산을 병렬로 수행하는 데에 우수한 성능을 보이며, CPU로부터 연산에 필요한 정보를 받아 수행한 후 다시 CPU로 연산한 결과를 보내기 때문에 추가적인 정보 전송 시간이 필요하다^{[6][7]}.

본 논문에서는, 스펙클 제거 필터의 처리 속도를 증가

시키기 위하여 GPU를 사용하였으며, 기존의 CPU에 최적화된 스펙클 제거 필터의 알고리즘을 GPU의 병렬화 연산에 맞게 최적화를 수행하였다. 한국에서 측정한 KOMPSAT-5 위성 SAR 영상을 기반으로 스펙클 제거 알고리즘을 수행하였으며, 기존의 알고리즘과 최적화된 알고리즘에 대하여 CPU와 GPU를 각각 적용하여 알고리즘 수행 시 필요한 수행시간과 데이터 저장공간을 분석하였다.

II. 기존의 스펙클 제거 필터 알고리즘

2-1 변화탐지에서 스펙클 제거의 중요성

SAR 영상에서 스펙클의 존재는 영상 내의 표적의 탐지 및 식별 성능에 영향을 줄 수 있는데, 대표적인 예시로 변화탐지 수행 시 탐지율 대비 오탐지율 성능을 크게 감소시킨다. 변화탐지란, 동일한 지역에 대하여 다른 시간대에 촬영한 두 영상을 비교하여 변화가 존재하는 지역 혹은 물체를 탐지하는 것이다. 그림 1은 영상 내에 표적의 위치변화가 존재하는 SAR 영상 쌍을 보여주며, 표기한 원 내부의 관심 표적이 시험 영상에서 나타나는 것을 확인할 수 있다. 지역의 변화 여부는 두 영상의 대수(log)비를 구하여 알 수 있다.

$$R = \log \left[\frac{|I_{test}|}{|I_{ref}|} \right] \quad (1)$$

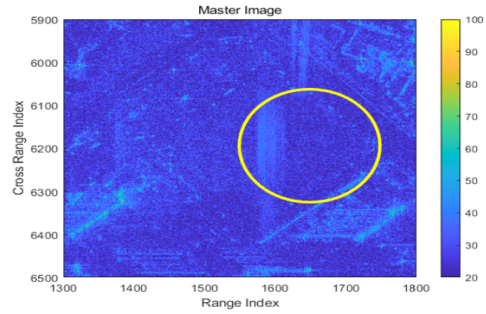
I_{test} 는 시험 SAR 영상이며, I_{ref} 는 비변화의 기준이 되는 SAR 영상을 의미한다. 식 (1)로부터 영상의 변화 정도가 클수록 R 영상의 화소 크기가 0에서 멀어진다.

그림 2는 SAR 영상 쌍으로부터 형성된 R 영상이며, 스펙클이 포함된 R 영상(a)과 스펙클 필터를 적용한 R 영상(b)을 보여준다. 이로부터 스펙클이 포함된다면 변화 표적이 스펙클에 묻히는 것을 확인할 수 있다.

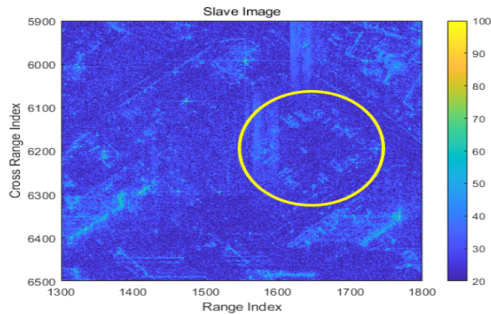
그림 3은 그림 2에 탐지 알고리즘을 수행한 뒤 탐지율 대비 오탐지율을 분석한 결과이다. 탐지율, P_D ,과 오탐지율, P_F ,은 다음과 같다.

$$P_D = \frac{N_D}{N} \quad (2)$$

$$P_F = \frac{N_F}{N - N_T} \quad (3)$$



(a) 참조 SAR 시뮬레이션 영상, I_{ref}
(a) Reference SAR simulation image, I_{ref}



(b) 시험 SAR 시뮬레이션 영상, I_{test}
(b) Test SAR simulation image, I_{test}

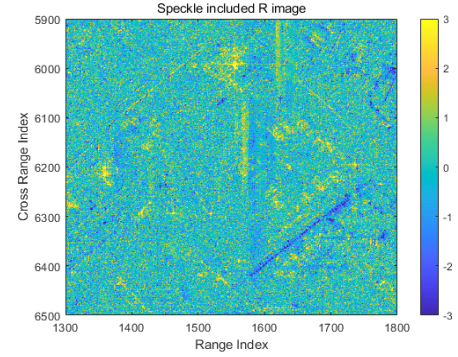
그림 1. 다중 시기 SAR 시뮬레이션 영상
Fig. 1. Mutli-temporal SAR simulation pair.

식 (3)에서 N , N_D , N_F , N_T 는 각각 전체 화소 수, 탐지된 화소 수, 오탐지된 화소 수, 표적 화소 수를 의미한다. 그림 3으로부터 스펙클이 영상에 포함될 경우, 변화 탐지 영상의 탐지율 대비 오탐지율 성능을 크게 떨어트리며, 이로 인해 SAR 영상에서 스펙클을 제거하는 것이 중요하다는 것을 확인할 수 있다.

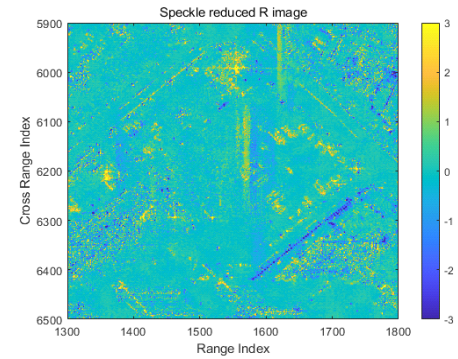
2-2 지역 필터(Local Filter)

지역 필터는 필터를 수행하려는 화소를 중심으로 이웃한 화소들의 강도를 참조하여 필터를 수행하며, 평균값 필터, 중앙값 필터, 리 필터, 향상된 리 필터 등이 있다.

그림 4는 일반적인 지역 필터의 구조를 보여준다. 영상 내에 슬라이딩 윈도우(sliding window)를 설정하고, 슬라이딩 윈도우 내부의 화소들, s_i ,의 값을 바탕으로 중심 화소의



(a) 스펙클이 포함된 R 영상
(a) Speckle included R image



(b) 스펙클이 감소된 R 영상
(b) Speckle reduced R image

그림 2. 생성된 R 영상
Fig. 2. R image from image pair.

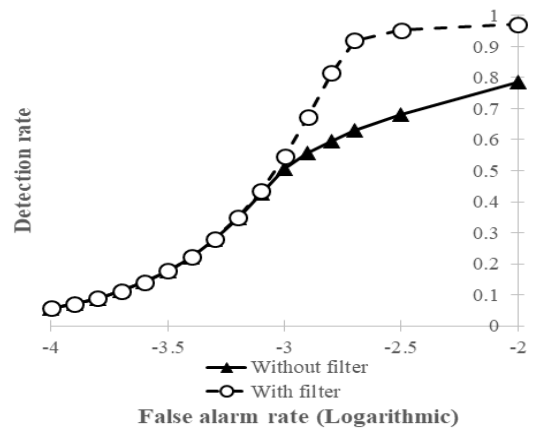


그림 3. 탐지율 대비 오탐지율
Fig. 3. Detection rate vs false alarm rate.

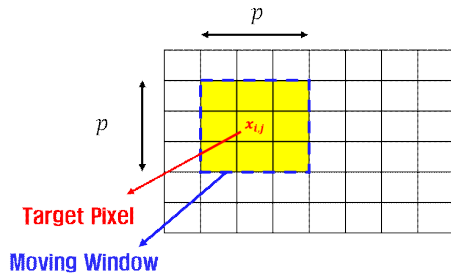


그림 4. 일반적인 지역 필터 구조
Fig. 4. The general structure of local filter.

값을 재설정한다. 평균값 필터, x_{mean} ,와 중앙값 필터, x_{median} ,는 각각 중심 화소의 값을 슬라이딩 윈도우 내 화소들의 강도 평균 및 중앙값으로 설정한다.

$$x_{mean} = \frac{1}{p^2} \sum_{i=1}^p s_i \quad (4)$$

$$x_{median} = median(s_i) \quad (5)$$

리 필터, x_{Lee} ,의 경우, 슬라이딩 윈도우 내의 화소들의 강도 값들을 바탕으로 문턱값을 계산하고, 문턱값보다 작은 화소들의 평균값으로 중심 화소의 크기를 설정한다.

$$x_{Lee} = \frac{\sum_{i=1}^p \delta_i s_i}{\sum_{i=1}^p \delta_i} \quad (6)$$

δ_i 는 문턱값보다 작은 화소를 선택하기 위한 논리값이며 아래의 식을 통해 결정한다.

$$\delta_i = \begin{cases} 1, & (1-2\sigma_s)\mu_s < s_i < (1+2\sigma_s)\mu_s \\ 0, & otherwise \end{cases} \quad (7)$$

식 (7)에서 μ_s 와 σ_s 는 각각 슬라이딩 윈도우 내 화소들의 평균과 표준편차를 의미한다. 마지막으로, 향상된 리 필터는 리 필터의 문제점인 실제 SAR 영상의 화소 강도 분포가 비대칭이라는 것과 높은 반사율을 가진 표적이 블러링(blurring) 되는 것을 해결한 필터이다. 화소 강도 분포가 비대칭인 점을 해결하기 위하여 새로운 문턱 값을 설정하였으며, 높은 반사율을 가진 표적이 블러링되는 문제는 필터를 수행하기 전 윈도우 내에 일정 비율 이상의 화

소 강도가 높을 경우, 표적으로 간주하여 필터를 거치지 않고 보존함으로써 해결하였다.

2-3 비지역 필터(Non-Local Filter)

비지역 필터는 필터를 수행하려는 화소를 중심으로 검색 윈도우(search window)를 설정하여, 검색 윈도우 내부의 모든 화소를 참조하여 필터를 수행한다.

그림 5는 대표적인 비지역 필터 중 하나인 비지역 평균 필터의 구조를 보여준다. 비지역 평균 필터는 이전의 지역 필터와 달리 패치 단위로 필터를 수행하며, 검색 윈도우 내부의 패치들을 참조하여 표적 패치, x_{NLM} 의 값을 설정한다.

$$x_{NLM} = \sum_{q_i \in S} \omega(p, q_i) \cdot q_i \quad (8)$$

p 는 필터를 수행하려는 표적 패치를, q_i 는 검색 윈도우 S 내에서 선택된 비교 패치를, $\omega(p, q_i)$ 는 가중치를 의미하며 가중치의 값은 표적 패치와 비교 패치의 유사도가 높을수록 큰 값을 할당하며, 유사도는 두 패치 간의 화소 강도의 유클리디안 거리(euclidean distance)를 이용하여 결정한다.

$$\omega(p, q_i) = \exp\left(-\frac{\|p - q_i\|_2^2}{h^2}\right) \quad (9)$$

h 는 지수 함수의 값이 수렴하는 속도를 결정하는 값이

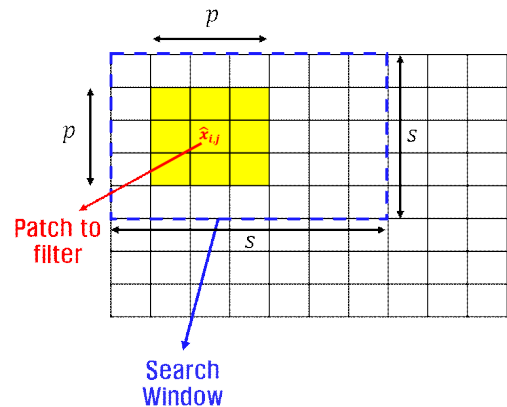


그림 5. 비지역 평균 필터의 구조
Fig. 5. The structure of non-local means filter.

다. 검색 창 크기가 클수록 참조하는 패치의 수가 증가하며, 스펙클 제거 및 에지 보존 성능은 향상되지만, 처리 시간이 증가하기 때문에 적절한 검색 창 크기를 설정하는 것이 중요하다.

2.4 스펙클 제거 필터의 성능분석

스펙클 제거 필터의 성능을 확인하기 위한 값으로는 effective number of looks (ENL), contrast, entropy가 있으며, ENL의 값이 클수록 영상의 스펙클 제거량이 많음을 의미한다^[8]. Contrast의 값이 크다면 영상의 에지 보존이 잘 되었음을 의미하며^[9], entropy의 값이 작다면 영상의 초점 보정이 잘 되었음을 의미한다^[9]. ENL과 contrast, C 및 entropy, ϵ 는 아래와 같이 구할 수 있다.

$$ENL(I[M,N]) = \frac{E(|I[M,N]|^2)}{\sigma[|I[M,N]|^2]} \quad (10)$$

$$C(I[M,N]) = \frac{\sigma[|I[M,N]|^2]}{E(|I[M,N]|^2)} \quad (11)$$

$$\epsilon(I[M,N]) = \sum \left(\frac{|I[M,N]|^2}{\sum |I[M,N]|^2} \ln \frac{\sum |I[M,N]|^2}{|I[M,N]|^2} \right) \quad (12)$$

위의 수식에서 E 는 평균을, σ 는 표준편차를 의미한다. 스펙클 제거 필터의 성능 수치와 연산 속도를 분석하기 위하여 10,000×10,000 크기의 SAR 영상에 대하여 슬라이딩 윈도우의 크기 p 는 3, 검색 창 크기 s 는 51로 설정하였다. 표 1은 각 스펙클 제거 필터를 수행한 후 각 필터의 성능 수치를 보여준다. 평균값, 중앙값 필터의 경우, 스펙클 제거, 에지 보존, 초점 성능이 떨어지지만 처리 속도가

표 1. 각 스펙클 제거 필터의 수행 성능

Table 1. The performance of speckle reduction filters.

| Filter name | Mean | Median | Lee | Enhanced lee | Non-local means |
|-----------------------|--------|--------|--------|--------------|-----------------|
| ENL | 17.476 | 12.403 | 13.109 | 58.149 | 738.28 |
| Contrast | 4.5900 | 5.1830 | 5.1140 | 8.9161 | 7.3633 |
| Entropy | 16.881 | 16.95 | 17.001 | 15.719 | 15.738 |
| Processing time [sec] | 0.488 | 0.593 | 5.852 | 8.081 | 825.68 |

매우 우수하며, 리 필터 및 향상된 리 필터는 평균값 혹은 중앙값 필터보다 상대적으로 우수한 성능을 보이지만 처리 시간이 상대적으로 길며, 비지역 평균 필터의 경우, 가장 우수한 필터 성능을 보이지만 처리 시간이 상당히 긴 것을 확인할 수 있다. 이를 통해 성능이 우수한 필터는 처리 시간이 오래 걸리는 문제점이 존재하며, 실시간 처리와 같이 필터의 처리 속도가 중요한 경우, 스펙클 제거 성능이 우수한 필터를 사용하기 어려운 것을 알 수 있다. 이를 개선하기 위하여 GPU 기반의 스펙클 제거 알고리즘을 제안한다.

III. GPU 기반 스펙클 제거 알고리즘 최적화

3.1 GPU 사용 시 기존의 직렬 알고리즘의 문제점

이전에 언급한 것처럼, GPU는 코어 하나의 성능이 떨어지는 대신 많은 수의 코어를 가지고 있어 단순한 연산을 병렬로 처리하는 것에 우수한 성능을 보인다.

그림 6은 MATLAB 2020A 환경에서 CPU 및 GPU를 사용하여 행렬 곱(matrix multiplication) 연산을 행렬의 크기를 증가시키면서 분석한 결과이다. 곱은 실선은 CPU를, 얇은 실선은 GPU의 연산 수행 속도를 의미하며, 높은 수치를 보일수록 처리 속도가 빠름을 의미한다. 그림 6으로부터 연산을 수행하는 원소 수가 적다면 CPU의 행렬 곱 수행 속도가 GPU와 유사하거나 오히려 더 빠른 것을 확인

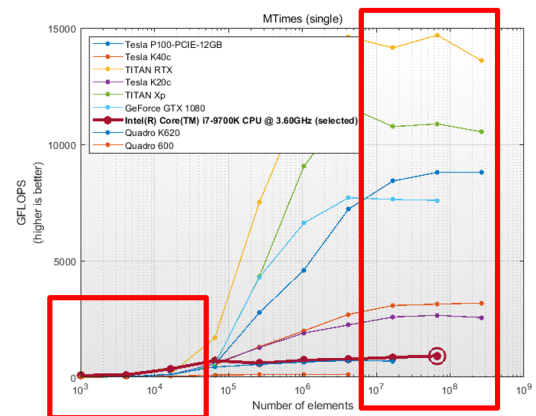


그림 6. CPU 및 GPU의 행렬 곱 연산 성능

Fig. 6. The matrix multiplication processing speed of CPU and GPU.

할 수 있으며, 반대로 연산을 수행하는 행렬의 원소 수가 많다면 GPU의 병렬 연산의 효율을 최대화할 수 있어 GPU의 연산 속도가 CPU와 비교하여 훨씬 빠른 것을 확인할 수 있다. 이를 통해 GPU의 병렬 연산 성능을 최대화하기 위해서는 한 번에 연산을 수행하는 행렬의 크기를 최대화해야 한다는 것을 알 수 있다. 하지만, 이전에 언급한 스펙클 제거 알고리즘들은 영상 내에 슬라이딩 윈도우를 설정한 뒤, 표적 화소의 값을 슬라이딩 윈도우 내부의 화소들을 참조하여 설정하기 때문에, 작은 크기의 연산은 많은 수를 반복하게 되어 GPU를 사용하기에 부적합하다.

3.2 스펙클 제거 알고리즘의 병렬 방식 최적화

스펙클 제거 알고리즘을 GPU의 병렬 연산에 최적화하기 위하여 행렬 연산의 크기를 최대화한다. 그림 7은 제안하는 알고리즘을 보여주며, 기존의 알고리즘과의 차이점은 슬라이딩 윈도우의 수가 $p \times p$ 만큼 생성되도록 크기를 조절한다. 이렇게 생성한 슬라이딩 윈도우를 재구성하여 그

림 7(b)와 같이 3차원 행렬을 구성한다. 이때, 재구성된 3차원 행렬의 3번째 축 방향의 원소들은 표적 화소를 필터링하기 위해 참조하려는 화소들로 구성되어 있음을 확인할 수 있다. 그러므로, 단순히 3차원 축 방향으로 연산을 수행한다면, 영상의 모든 화소에 대하여 동시에 필터를 수행할 수 있다. 만약 슬라이딩 윈도우를 사용하여 잘라낸 윈도우를 S_i 라고 한다면, 평균값 혹은 중앙값 필터와 같은 단순한 구조의 지역 필터는 다음 수식을 통해 구할 수 있다.

$$I_{mean} = \frac{1}{p^2} \sum_{i=1}^{p^2} S_i \quad (13)$$

$$I_{median} = median(S_1, S_2, \dots, S_{p^2}) \quad (14)$$

지역 필터 중 복잡한 알고리즘을 지닌 리 필터의 경우, 아래와 같이 구할 수 있다.

$$I_{Lee} = \frac{\sum_{i=1}^{p^2} \delta_i S_i}{\sum_{i=1}^{p^2} \delta_i} \quad (15)$$

리 필터 알고리즘에서 평균값을 구하는 데 사용할 화소를 선택하는 논리값인 δ_i 는 다음과 같이 모든 화소에 대하여 동시에 구할 수 있다.

$$\delta_i = \begin{cases} 1, & (1 - 2\sigma(S))\mu(S) \leq S_i \leq (1 + 2\sigma(S))\mu(S) \\ 0, & otherwise \end{cases} \quad (16)$$

$\sigma(S)$ 와 $\mu(S)$ 는 재구성한 3차원 행렬의 3차원 축 방향 원소들의 평균 및 표준편차로 이루어진 2차원 행렬이다.

$$\mu(S) = \frac{1}{p^2} \sum_{i=1}^{p^2} S_i \quad (17)$$

$$\sigma(S) = \frac{1}{p^2} \sum_{i=1}^{p^2} \sqrt{\frac{1}{p^2} \sum_{i=1}^{p^2} (\mu(S) - S_i)^2} \quad (18)$$

향상된 리 필터는 리 필터의 구조를 동일하게 사용하며, 차이점은 문턱 범위를 새롭게 설정한다는 것과 필터를 수행하기 전 고 반사율을 지닌 표적을 선택하는 점이다. 그러므로 최적화된 병렬 방식의 알고리즘을 동일하게 적용할 수 있다.

병렬 방식의 알고리즘은 비지역 평균 필터에 대해서도

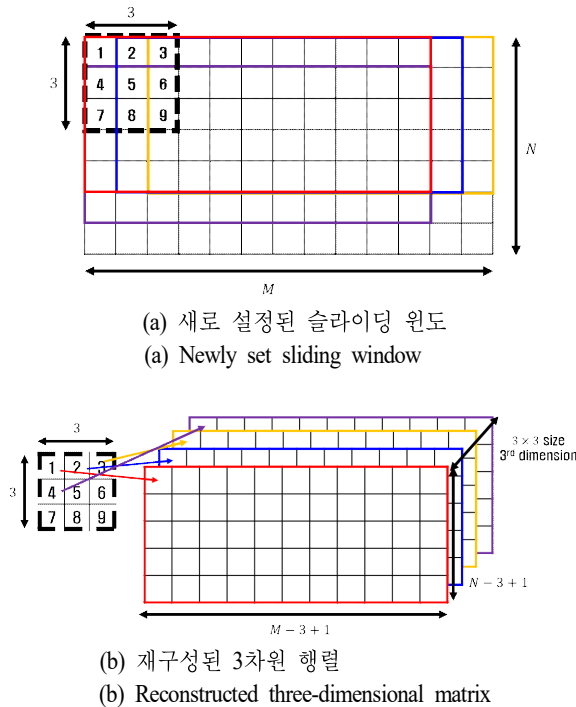


그림 7. 제안하는 스펙클 제거 알고리즘, $p=3$
Fig. 7. Proposed speckle reduction algorithm, $p=3$.

슬라이딩 윈도우를 구성하여 필터를 수행하기 때문에 동일하게 적용할 수 있다. 이전의 지역 필터들은 화소 단위로 필터를 수행하였지만, 비지역 평균 필터의 경우 패치 단위로 수행하기 때문에, 슬라이딩 윈도우의 수가 검색 창 내부의 화소의 수와 일치하도록 구성한다면 영상 내의 모든 패치에 대해서 동시에 필터 연산을 수행할 수 있다.

지역 필터의 기존의 알고리즘이 $O(p \times p)$ 크기의 연산을 $M \times N$ 횟수만큼 반복해야 한다면, 제안한 알고리즘은 $O(M \times N)$ 크기의 연산을 $p \times p$ 횟수만큼 반복한다. 또한, 비지역 평균 필터의 경우, 기존의 알고리즘이 $O(p \times p)$ 크기의 연산을 $s \times s \times M \times N$ 횟수만큼 반복해야 한다면, 제안한 알고리즘은 $O((M \times N) \times (p \times p))$ 크기의 연산을 $s \times s$ 횟수만큼 반복한다. 즉, 제안한 알고리즘은 전체 연산량은 유지하지만, 대량의 연산을 적은 수만큼 반복하도록 변경하여 GPU를 사용하기에 적합한 구조를 가진다.

IV. SAR 영상 시뮬레이션 및 분석결과

4.1 시뮬레이션 환경

본 논문에서 GPU에 최적화된 알고리즘을 분석하기 위해 설정한 시뮬레이션 환경은 표 1과 같다.

SAR 영상에 대하여 스펙클 제거 필터의 직렬 방식 알고리즘과 병렬 방식 알고리즘을 각각 CPU를 적용하였을 때와 GPU를 적용하였을 때를 비교 및 분석하였으며, 기존의 알고리즘과 제안한 알고리즘을 수행할 때 필요로 하는 최소 저장공간을 비교 및 분석하였다. 또한, 스펙클 제거를 수행하는 SAR 영상의 크기, $M \times N$,는 1,250×1,250, 2,500×2,500, 5,000×5,000, 10,000×10,000으로 수행하였고, 지역 필터를 수행할 때 설정하는 참조 영역 크기, $p \times p$,는 3×3, 5×5, 7×7, 9×9로 수행하였으며, 비지역 평균 필터를

수행할 때 필요한 검색 창의 크기, $s \times s$,는 51×51, 101×101, 201×201, 401×401로 수행하였다. 필터를 수행할 때 하나의 값만을 변경하며 수행하였으며, 기본값으로 SAR 영상의 크기는 10,000×10,000, 참조 영역 크기는 3×3, 검색 창의 크기는 51×51로 설정하였다.

4.2 시뮬레이션 결과

먼저, 지역 필터인 평균값, 중앙값, 리, 향상된 리 필터에 대하여 시뮬레이션을 수행한 결과는 그림 8에서 확인할 수 있다. 단순한 구조의 필터인 평균값 혹은 중앙값 필터의 경우, 제안한 알고리즘 사용 시 행렬을 재구성하는데 필요한 시간이 병렬 연산을 통해 단축된 시간보다 크기 때문에, 오히려 제안한 알고리즘의 처리 시간이 증가한 것을 확인할 수 있다. 하지만, 리 필터와 향상된 리 필터와 같이 복잡한 필터의 경우, 행렬을 재구성하는데 필요한 시간이 병렬 연산을 수행하여 단축된 시간보다 훨씬 적어 처리 시간이 크게 단축된 것을 확인할 수 있다. 또한, 기존의 알고리즘에 GPU를 적용하였을 때와 제안한 알고리즘에 CPU를 적용하였을 때 연산 효율이 떨어져 상대적으로 긴 처리 시간을 확인할 수 있다.

다음으로, 그림 9는 비지역 평균 필터에 대하여 시뮬레이션을 수행한 결과이다. 병렬 방식을 적용할 경우, 수행하는 연산의 크기가 지역 필터에 병렬 방식을 적용했을 때보다 더 증가하기 때문에, GPU를 사용하였을 때 제안한 방식의 처리 시간이 지역 필터보다 크게 단축되는 것을 확인할 수 있다. 또한, 직렬 방식의 알고리즘에 GPU를 사용하였을 때와 병렬 방식의 알고리즘에 CPU를 사용하였을 때 연산장치의 효율이 떨어져 처리 시간이 크게 증가하는 것을 확인할 수 있다.

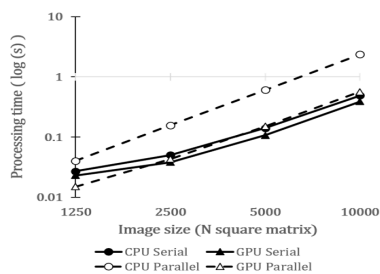
표 3은 스펙클 제거 필터들에 대하여 기존의 직렬 방식의 알고리즘과 병렬 방식 알고리즘을 수행하는 데에 필요한 최소 저장공간을 보여준다. 병렬 방식 알고리즘은 전체 영상의 화소에 대하여 필터 연산을 동시에 수행하기 때문에, 수행하는 데 필요한 저장공간이 직렬 방식보다 훨씬 더 필요한 것을 확인할 수 있다.

결과적으로, GPU를 사용한 병렬 방식 알고리즘은 영상의 품질을 유지하면서 복잡한 구조의 스펙클 제거 필

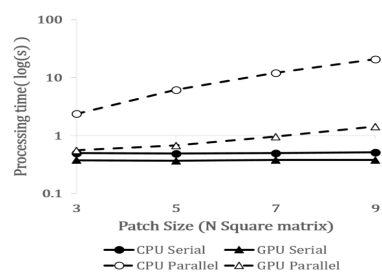
표 2. 시뮬레이션 환경

Table 2. Simulation environment.

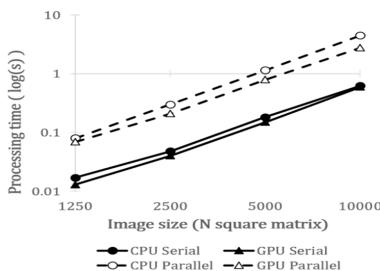
| | |
|--------------|---------------------|
| Software | MATLAB 2020A |
| CPU | Intel Core i7-9700K |
| GPU | NVIDIA TITAN RTX |
| CPU capacity | 64 [GB] |
| GPU capacity | 24 [GB] |



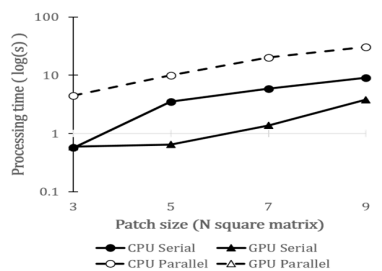
(a) 영상 크기에 따른 평균값 필터 성능
(a) Processing time of mean filter compared to image size



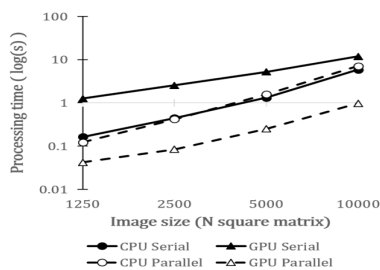
(b) 참조 영역 크기에 따른 평균값 필터 성능
(b) Processing time of mean filter compared to patch size



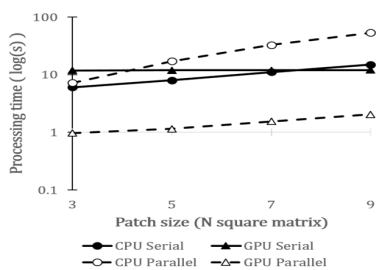
(c) 영상 크기에 따른 중앙값 필터 성능
(c) Processing time of median filter compared to image size



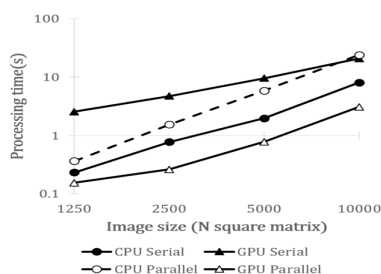
(d) 참조 영역 크기에 따른 중앙값 필터 성능
(d) Processing time of median filter compared to patch size



(e) 영상 크기에 따른 리 필터 성능
(e) Processing time of Lee filter compared to image size



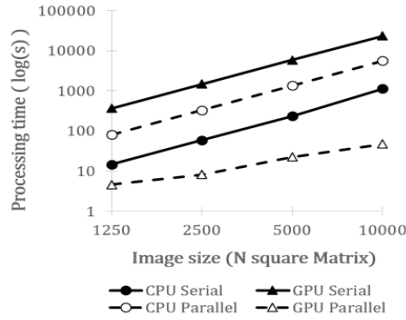
(f) 참조 영역 크기에 따른 리 필터 성능
(f) Processing time of Lee filter compared to patch size



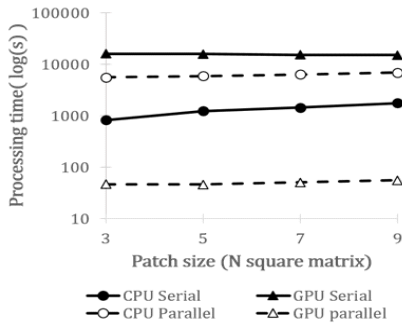
(g) 영상 크기에 따른 향상된 리 필터 성능
(g) Processing time of enhanced Lee filter compared to image size

그림 8. SAR 영상에 지역 필터를 적용했을 때 처리 속도 분석 결과

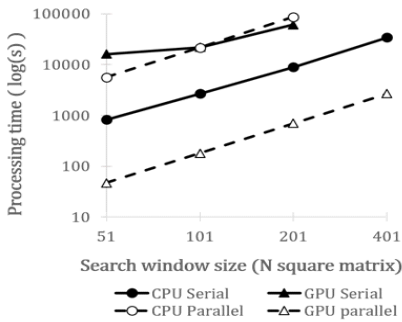
Fig. 8. The processing speed analysis of local filters applied to SAR image.



(a) 영상 크기에 따른 비지역 평균 필터 성능
(a) Processing time of NL-means filter compared to image size



(b) 패치 크기에 따른 비지역 평균 필터 성능
(b) Processing time of NL-means filter compared to patch size



(c) 검색 영역 크기에 따른 비지역 평균 필터 성능
(c) Processing time of NL-means filter compared to search window size

그림 9. SAR 영상에 비지역 평균 필터를 적용했을 때 처리 속도 분석 결과

Fig. 9. The processing speed analysis of non-local means filter applied to SAR image.

터의 처리 속도를 향상시키는 데 유리하다. 하지만, 단순 구조의 필터는 오히려 처리 속도를 감소시키며, 영상의

표 3. 스펙클 제거 필터의 최소 필요 저장공간

Table 3. Minimum data capacity of speckle reduction filters.

| Filter name | Minimum data capacity [MB] | |
|-----------------|----------------------------|----------|
| | Serial | Parallel |
| Mean | 400 | 800 |
| Median | 400 | 3,600 |
| Lee | 400 | 1,600 |
| Enhanced Lee | 400 | 2,000 |
| Non-local means | 800 | 2,400 |

모든 화소에 대하여 동시에 연산을 수행하기 때문에, 알고리즘을 수행하기 위하여 직렬 방식의 알고리즘보다 더 많은 저장공간을 요구하며, 이는 GPU의 적은 저장공간 특성 때문에 가속화를 위해 병렬 구조로 처리 가능한 영상의 크기가 직렬 구조보다 작은 한계점이 존재한다.

V. 결 론

본 논문에서는 SAR 영상에 포함되는 스펙클을 제거하는 필터의 처리 속도를 향상시키기 위하여 GPU에 병렬 방식의 알고리즘을 사용하여 최적화하였으며, 제한한 방식을 사용하였을 때 처리 시간의 감소 수준과 필요한 최소 데이터 저장공간을 분석하였다. 시뮬레이션을 위하여 스펙클이 포함된 SAR 영상을 준비한 뒤, 스펙클 제거 필터에 대해 기존의 방식과 제한한 방식에 대하여 CPU 및 GPU를 각각 사용하였을 때 처리 시간을 분석하였다. 병렬 방식에 GPU를 사용하였을 경우, 필요한 저장 공간이 크게 증가하는 문제점이 존재하지만, 복잡한 구조를 가진 스펙클 제거 필터의 처리 시간을 크게 단축할 수 있었다. 이를 통해, 기존에는 실시간 탐지와 같은 처리 시간이 중요한 상황에서 스펙클 제거 성능이 우수한 필터를 사용하지 못하였지만, GPU를 사용한다면 스펙클 제거 성능이 우수한 필터 또한 고려할 수 있을 것으로 기대된다.

References

- [1] M. Forouzanfar, H. Abrishami-Moghaddam, "Ultrasound speckle reduction in the complex wavelet domain," in

- Principles of Waveform Diversity and Design*, M. Wicks, E. Mokole, S. Blunt, R. Schneible, and V. Amuso, Eds. Raleigh, NC, SciTech, 2010, pp. 558-577.
- [2] A. Fabrizio, A. Lapini, T. Bianchi, and L. Alparone, "A tutorial on speckle reduction in synthetic aperture radar images," *IEEE Geoscience and Remote Sensing Magazine*, vol. 1, no. 3, pp. 6-35, Sep. 2013.
- [3] J. S. Lee, "A simple speckle smoothing algorithm for synthetic aperture radar images," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-13, no. 1, pp. 85-89, Jan.-Feb. 1983.
- [4] J. S. Lee, J. H. Wen, T. L. Ainsworth, K. S. Chen, and A. J. Chen, "Improved sigma filter for speckle filtering of SAR imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 1, pp. 202-213, Jan. 2009.
- [5] B. Antoni, B. Coll, and J. M. Morel, "A non-local algorithm for image denoising," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition(CVPR'05)*, San Diego, CA, Jun. 2005, vol. 2, pp. 60-65.
- [6] Y. H. Lee, H. Chin, and K. T. Kim, "Acceleration of the iterative physical optics using graphic processing unit," *The Journal of Korean Institute of Electromagnetic Engineering and Science*, vol. 26, no. 11, pp. 1012-1109, Nov. 2015.
- [7] Mathworks, *MATLAB Version 9.8.0.145132 (R2020a)*, Natick, MA, 2020.
- [8] J. S. Lee, P. Eric, *Polarimetric Radar Imaging: From Basics to Applications*, Boca Raton, FL, CRC Press, 2017.
- [9] J. Wang, X. Liu, and Z. Zhou, "Minimum-entropy phase adjustment for ISAR," *IEE Proceedings-Radar, Sonar and Navigation*, vol. 151, no. 4, pp. 203-209, Aug. 2004.

이 건 [포항공과대학교/박사과정]

<https://orcid.org/0000-0001-6607-5177>



2019년 8월: 경북대학교 전자공학부 (공학사)
 2021년 8월: 포항공과대학교 전자전기공학(공학석사)
 2021년 9월 ~ 현재: 포항공과대학교 전자전기공학과 박사과정
 [주 관심분야] 레이더 신호처리

이 명 준 [한국항공우주연구원/선임연구원]

<https://orcid.org/0000-0002-8073-2556>



2014년 8월: 한동대학교 전산전자공학부 (공학사)
 2017년 2월: 포항공과대학교 전자전기공학과 (공학석사)
 2017년 3월 ~ 2021년 8월: 포항공과대학교 전자전기공학과 (공학박사)
 2021년 8월 ~ 현재: 한국항공우주연구원

국가위성정보 활용지원센터 선임연구원
 [주 관심분야] 레이더 신호처리, SAR 표적 탐지, SAR 이동 표적 탐지, SAR 변화 탐지

정 남 훈 [포항공과대학교/박사과정]

<https://orcid.org/0000-0003-1629-2418>



2015년 8월: 포항공과대학교 전자전기공학과 (공학사)
2018년 2월: 포항공과대학교 전자전기공학과 (공학석사)
2018년 3월~현재: 포항공과대학교 전자전기공학과 박사과정
[주 관심분야] 레이더 신호 처리 및 영상,

레이더 자원관리

김 경 태 [포항공과대학교/교수]

<https://orcid.org/0000-0003-1200-5282>



1994년 2월: 포항공과대학교 전자전기공학과 (공학사)
1996년 2월: 포항공과대학교 전자전기공학과 (공학석사)
1999년 2월: 포항공과대학교 전자전기공학과 (공학박사)
2002년 3월~2011년 2월: 영남대학교 전

자공학과 교수

2011년 3월~현재: 포항공과대학교 전자전기공학과 교수
2012년 9월~2017년 12월: 레이더/IR 표적식별 특화연구실실장
2018년 1월~현재: 무인감시정찰기술연구센터장
2019년 4월~현재: 차세대 영상레이더시스템 연구센터장
2020년 11월~현재: 차세대 국방융합기술 연구센터장
[주 관심분야] 레이더 신호 처리 및 영상, 레이더 표적인식 및 패턴인식, 전자기수치해석 및 RCS 측정

최 재 호 [포항공과대학교/박사과정]

<https://orcid.org/0000-0001-9484-4869>



2017년 2월: 고려대학교 컴퓨터·통신공학부 (공학사)
2019년 2월: 포항공과대학교 전자전기공학과 (공학석사)
2019년 3월~현재: 포항공과대학교 전자전기공학과 박사과정
[주 관심분야] 레이더 신호처리 및 영상,

휴먼 탐지, 미세도플러 분석