

# 평행사변형 분류 알고리즘의 성능에 대한 연구

용 환 기<sup>1</sup>

## A Study on the Performance of Parallelepiped Classification Algorithm

Whan-Ki YONG<sup>1</sup>

### 요 약

위성영상은 GIS 정보획득을 위한 가장 중요한 초기자료로서, 이로부터 주제도와 같은 유용한 정보를 추출하기 위해서는 위성영상 즉 다중스펙트럼 영상을 목적에 적합하게 분류하는 처리과정이 필요하다. 위성영상의 분류기법은 크게 감독기법과 무감독기법으로 나뉘는데, 본 논문에서는 감독분류기법 중의 하나인 평행사변형 알고리즘에서 군집의 초기값 설정이 알고리즘의 성능에 미치는 영향을 분석한다. 본 연구에서는 우선 직렬컴퓨터에서 평행사변형 알고리즘의 성능과 초기값 변화와의 관계를 살펴보고, 이를 확장하여 MIMD 병렬구조 컴퓨터 모델을 사용한 경우에 초기값의 변화가 평행사변형 알고리즘의 성능에 미치는 영향을 분석한다. 평행사변형 알고리즘의 성능은 초기값의 설정에 따라 직렬구조의 컴퓨터를 사용하는 경우에는 최고 2.4배, 그리고 MIMD 병렬구조 모델을 사용한 경우에는 최고 2.5배의 성능 향상을 보였다. 전산모의실험을 통해 위성영상의 감독분류기법에서 초기값이 평행사변형 분류알고리즘의 성능에 상당한 영향을 미치며, 직렬컴퓨터와 MIMD 병렬컴퓨터에서 초기값의 적절한 설정을 통해 분류기법의 성능이 향상됨을 확인하였다.

주요어: 위성영상, 감독분류기법, 평행사변형 알고리즘, MIMD 병렬 모델

### ABSTRACT

Remotely sensed data is the most fundamental data in acquiring the GIS informations, and may be analyzed to extract useful thematic information. Multi-spectral classification is one of the most often used methods of information extraction. The actual multi-spectral classification may be performed using either supervised or unsupervised approaches. This paper analyze the effect of assigning clever initial values to image classes on the performance of parallelepiped classification algorithm, which is one of the supervised classification algorithms. First, we investigate the effect on serial computing model, then expand it on MIMD(Multiple Instruction Multiple Data) parallel computing model. On serial computing model, the performance of the parallel pipe algorithm improved 2.4 times at most and, on MIMD parallel computing model the performance improved about 2.5 times as clever initial values are

2001년 11월 12일 접수 Received on November 12, 2001

<sup>1</sup> 한동대학교 전산전자공학부 (wyong@handong.edu)

School of Computer Science and Electronics Engineering, Handong University

assigned to image class. Through computer simulation we find that initial values of image class greatly affect the performance of parallelepiped classification algorithms, and it can be improved greatly when classes on both serial computing model and MIMD parallel computation model.

**KEYWORDS:** Remote-sensed Data, Supervised Classification, Parallelepiped Algorithm, MIMD Parallel Computing Model

## 서 론

위성영상으로부터 의미있는 정보를 담고 있는 주제도(thematic map)를 작성하기 위해서는 영상의 화소를 비슷한 성질을 가지는 군집들로 분류하는 과정이 필수적이다. 분류과정을 통해 만들어진 주제도는 GIS의 하나의 자료층으로 이용될 수 있는 자료형태를 가진다.

이를 위한 분류기법으로는 크게 감독분류와 무감독분류로 나누어진다(Jenson, 1996). 감독분류는 분류하고자 하는 자료들에 대한 선험적 지식(prior knowledge) 즉, 분류할 항목의 개수, 분류할 항목들의 군집중심, 밴드간 공분산 등과 같은 통계적 특성 등을 이용하여 자료들을 분류하는 방법이다. 감독분류의 대표적인 알고리즘으로는 최소거리(minimum distance) 알고리즘과 평행사변형(parallelepiped) 알고리즘, 최대유사(maximum likelihood) 알고리즘 등이 있다. 무감독 분류는 자료들에 대한 선험적 지식없이 자료들만을 입력으로 받아들여 자료들을 군집화과정을 통해 각 분류항목들의 특성을 찾아내어 분류하는 기법으로서, 대표적인 알고리즘으로는 K-Means 알고리즘, ISODATA 알고리즘, 신경회로망을 이용하여 분류하는 방법 등이 있다. 최근에는 무감독분류를 통하여 자료를 일차적으로 군집화한 후, 전문가가 별도의 정보를 사용하여 군집화된 결과를 분석, 편집하여 분류항목별 이름, 군집중심, 밴드간 공분산 등을 새롭게 설정한 후 최종적으로 분류를 수행하는 복합(hybrid) 분류기법이 보다 널리 이용되고 있다.

평행사변형 알고리즘은 감독분류기법 중의 하나로서, 그 분류방법은 각 군집의 평균과 표

준편차의 합과 차로 생기는 평행사변형 내에 화소가 존재할 경우 그 군집에 속하는 것으로 분류하는 방법이다. 수식으로 간단히 나타내면 식 (1)과 같다.

$$z_{jk} - \sigma_{jk} \leq x_k \leq z_{jk} + \sigma_{jk} \quad \forall k, k=1,2,3,\dots,n \quad (1)$$

여기서,  $k$ 는 분류할 픽셀의 밴드 수이며  $z_j$ 와  $\sigma_j$ 는  $j$ 군집의 평균과 표준편차이다. 평행사변형 알고리즘은 계산량이 많지 않아 빠르게 분류하는 장점이 있으나 분류하지 못하는 화소가 존재하는 단점이 있다.

본 논문에서는 군집의 초기값 설정이 평행사변형 알고리즘의 성능에 미치는 영향을 살펴보기 위해서 직렬구조의 컴퓨터와 병렬구조의 컴퓨터 각각에 대해 전산모의실험을 수행한다.

## 직렬 컴퓨터에서 군집 초기값이 평행사변형 알고리즘에 미치는 영향

### 1. 평행사변형 알고리즘의 개선

평행사변형 알고리즘은 각 화소의 밝기값이 비교하고자 하는 군집의 평균 밝기값을 중심으로 한 일정 범위 내에 속하면 그 군집을 해당 화소의 군집으로 결정한다. 이때, 임의의 화소에 대하여 어느 군집을 먼저 검사하느냐에 따라 알고리즘의 성능은 차이가 난다(Bryant, 1979; 1989). 즉, 이웃 화소의 군집을 자신의 초기 군집로 정하여 이 군집부터 검사를 할 경우 화소가 이 군집에 귀속될 확률이 높아지므로 군집을 검사하는 loop이 조기에 탈출되어 계산 시간을 줄일 수 있다. 그림 1은

일반적인 평행사변형 알고리즘이고, 그림 2는 성능향상을 위하여 개선된 평행사변형 알고리즘의 pseudo코드이다.

```

for each pixel(i,j)
  for class l=0 to c-1
    index[i,j] = -1
    for k=1 to band
      if ( (mean[l,k]-sdev[l,k])<=intensity[i,j,k]
          && (intensity[i,j,k]<=(mean[l,k]+sdev[l,k]) )
        index[i,j] = l
    exit enclosing for loop
    
```

그림 1. 일반적인 평행사변형 알고리즘

```

init = 0
for each pixel(i,j)
  l = init
  for class l=init, init+1, init+2..., c-1, 0, ..., init-1
    index[i,j] = -1
    for k=1 to band
      (1) if ( (mean[l,k]-sdev[l,k])<=intensity[i,j,k]
          && (intensity[i,j,k]<=(mean[l,k]+sdev[l,k]) )
        index[i,j] = l
        init = l
    exit enclosing for loop
    
```

그림 2. 개선된 평행사변형 알고리즘

위의 알고리즘에서 변수 *index*는 각 화소가 속할 군집, 변수 *intensity*는 각 화소의 값, 그리고 변수 *mean*과 *sdev*는 각각 평균과 표준편차를 나타낸다. 그림 2의 코드와 그림 1의 코드의 차이점은 군집을 결정하는 loop이 그림 1의 코드에서는 군집 0부터 *c*-1까지 수행되지만, 그림 2의 코드에서는 바로 전에 군집이 정해진 이웃 화소의 군집(*init*)부터 군집을 검사하는 것이다. 그 이유는 현재 검사하고 있는 화소는 이웃에 있는 화소와 같은 군집에 속할 확률이 높기 때문이다.

## 2. 결과

그림 1과 그림 2의 pseudo코드를 C 언어로 구현하여 변형된 알고리즘의 성능 향상 정도를 살펴본다. 데이터는 포항시와 영일만 지역의 512×512 위성영상을 사용하였다(그림 3). 이 데이터 위성영상에는 17개의 군집이 존재한다.



그림 3. 포항시 영일만 위성영상

표 1은 일반적인 평행사변형 알고리즘과 개선된 평행사변형 알고리즘의 성능을 비교한 것이다. 군집의 범위를 정하는 문턱값으로는 1배 분산값, 2배 분산값, 그리고 3배 분산값의 세 가지 경우를 적용하였다. 표에서 분류율은 각 문턱값을 사용한 경우 전체 화소 중 군집으로 분류되는 화소의 비율을 나타낸다. 알고리즘의 성능을 나타내기 위해 일반 평행사변형 알고리즘에서 1배 분산값을 문턱값으로 사용한 경우의 loop count를 100으로 표시하여 기준으로 삼고, 나머지 경우의 상대적인 loop count를 표시하였다. 논문에서는 loop count를 계산할 때 그림 2의 코드 중 (1)번 구문이 수행되는 횟수를 기준으로 하였다.

**표 1.** 개선된 평행사변형 알고리즘의 성능 비교  
(단위: 100 = 25,800,355 loop count)

분산값	분류율(%)	일반	개선된
		평행사변형 알고리즘	평행사변형 알고리즘
1	24.5	100	96.6
2	87.9	51.3	33.6
3	98.6	35.5	14.7

표 1에서 살펴보면, 문턱값으로 1배의 분산값을 사용하면 분류율이 너무 낮고, 2배 혹은 3배의 분산값을 사용해야 만족할만한 분류가 이루어짐을 알 수 있다. 이 경우 이웃 화소의 군집을 자신의 초기 군집으로 하는 개선 알고리즘의 경우, 기존의 알고리즘에 비해 loop count의 수가 약 2/3와 5/12로 줄어드는 성능향상을 확인할 수 있다. 또, 분류율이 높을수록 성능향상의 정도가 더 커지는데, 이는 미분류 화소가 존재하는 경우에는 모든 군집에 대하여 화소의 귀속 여부를 검사하기 때문에 많은 계산이 요구되기 때문이다.

## 병렬 컴퓨터에서 군집 초기값이 평행사변형 알고리즘에 미치는 영향

### 1. 연산모델

본 논문에서는 병렬 컴퓨터의 대표적인 두가지 모델인 SIMD(Single Instruction Multiple Data) 및 MIMD(Multiple Instruction Multiple Data) 컴퓨터 모델 중 MIMD 모델에 평행사변형 알고리즘을 적용하였다.

SIMD 컴퓨터 모델에서 모든 프로세서는 같은 성능을 가진 동일한 프로세서들로 이루어지며, 각 프로세서는 각각의 제어유닛을 갖지 않고 중앙에 있는 한 개의 제어유닛이 전체 프로세서를 통제한다. 따라서 프로세서는 프로그램을 갖고 있지 않으며, 프로그램은 중앙의 제어유닛에 의하여 각 프로세서에 명령

어의 형태로 공급된다(Almasi 등 1994). 반면에 MIMD 컴퓨터 모델에서 각 프로세서는 제어유닛과 프로그램 그리고 데이터를 갖고 있다. 즉, 독립적인 여러 개의 컴퓨터들이 연결되어 있는 형태이다. 따라서, 각 프로세서는 독립적으로 다른 연산을 할 수 있다. 즉 한 프로세서가 곱셈 연산을 하는 동안 다른 프로세서는 덧셈 연산을 할 수 있다. 수행될 작업은 컴파일러 등에 의하여 여러개의 작은 작업으로 나뉘어져서 각 프로세서들로 공급된다. 각 프로세서들은 독립적으로 일을 수행하며, 필요에 따라 다른 프로세서들과 데이터를 교환한다. 프로세서간의 데이터 통신은 각 프로세서가 자신의 메모리를 갖고 메시지를 주고 받는 message passing 방법과 공통의 메모리를 두고 이 메모리를 통하여 데이터를 주고 받는 shared memory 방법이 있다(Hwang과 Briggs, 1984). 본 논문에서는 message passing 방법을 사용하는 것으로 가정한다.

우리는 여기서, 각 프로세서간의 통신 채널의 크기가 병렬 컴퓨터 word의 크기라고 가정하는 word 모델을 사용한다. 즉, 병렬 컴퓨터의 word의 크기가  $w$  bits 이라고 가정하면  $w$  bits의 데이터가  $O(1)$ 의 시간에 하나의 프로세서에서 이웃 프로세서로 이동할 수 있다. 만일 각 프로세서간의 통신 채널의 크기가 1 bit이라고 가정하는 bit 모델을 사용하면, 각 프로세서간의 통신 overhead는  $w$ 배 만큼 증가하고, 이는 알고리즘의 성능을 저하시킨다. 그러나, MasPar 등의 병렬컴퓨터는 각 프로세서간의 통신 채널의 크기가 여러 bits 이므로, word 모델을 사용하는 것은 합당하다.

위성영상의 크기를  $n \times n$  이라고 할 때, 이를 다루기 위해 그림 4에서와 같이  $n$ 개의 프로세서가 일렬로 연결되어 있는 일렬 배열형을 생각한다. 맨 위쪽의 프로세서를  $PE(0)$  이라 하고, 맨 아래쪽의 프로세서를  $PE(n-1)$  이라 한다. 경계에 있는 프로세서를 제외하면, 각 프로세서의 데이터 교환은 자신의 위쪽 혹

은 아래쪽에 위치한 프로세서와 이루어진다. 그림 4에서와 같이  $i=0$ 의 열에 있는  $n$ 개의 화소는  $PE(0)$ 에,  $i=p$ 에 있는  $n$ 개의 화소는  $PE(p)$ 에 mapping된다.

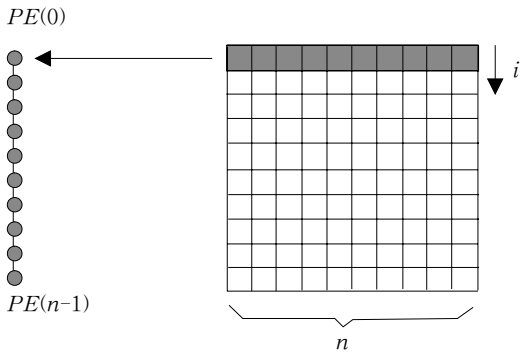


그림 4. 일렬 배열형 모델과  $n \times n$  화소의 mapping

## 2. 일반 알고리즘의 적용

그림 5는 평행사변형 알고리즘을 MIMD 모델에 적용시켰을 때의 pseudocode이다. SIMD 모델과는 달리 MIMD 모델에서는 화소의 군집이 정해졌을 때, 나머지 군집에 대한 'for' loop을 탈출하여 다음 프로그램을 수행하므로, 각 프로세서는 독립적으로 자신에 할당된 화소들에 대한 분류를 해 나갈 수 있다. 따라서, 이로 인한 많은 성능 향상을 기대할 수 있다.

```

for each PE(x,y)
  for each pixel(i,j)
    for class l=0 to c-1
      index[i,j] = -1
      for k=1 to band
        if ( (mean[l,k]-sdev[l,k])<=intensity[i,j,k])
          && (intensity[i,j,k]<=(mean[l,k]+sdev[l,k]) )
            index[i,j] = l
            exit enclosing for loop
    
```

그림 5. MIMD 모델에서의 평행사변형 알고리즘

## 3. 개선 알고리즘의 적용

그림 6은 평행사변형 알고리즘의 개선된 알고리즘을 병렬 컴퓨터 모델에 적용시켰을 때의 pseudo-code이다. 위의 경우와 마찬가지로, MIMD 모델의 경우는 다른 프로세서의 상태와 관계없이 프로그램이 진행되므로 많은 성능의 향상을 기대할 수 있다

```

for each PE(x,y)
  init = 0
  for each pixel(i,j)
    l = init
    for class l=init, init+1, init+2, ..., c-1, 0, ..., init-1
      index[i,j] = -1
      for k=1 to band
        (1) if ( (mean[l,k]-sdev[l,k])<=intensity[i,j,k])
              && (intensity[i,j,k]<=(mean[l,k]+sdev[l,k]) )
                index[i,j] = l
                init = l
            exit enclosing for loop
    
```

그림 6. MIMD 모델에서의 개선 평행사변형 알고리즘

## 4. 결과

표 2는 같은 실험 데이터를 사용했을 때, MIMD 모델에서 일반 평행사변형 알고리즘과 개선된 평행사변형 알고리즘의 성능을 비교한 것이다. 앞서 지적한 바와 같이 MIMD 모델에서는 각 프로세서가 'for' loop을 조기에 탈출하여 많은 성능의 향상을 이루었다.

표 3은 직렬 컴퓨터에서 일반 평행사변형 알고리즘의 loop count를 100으로 하였을 때 직렬 컴퓨터에서의 개선 알고리즘과 병렬 컴퓨터에서의 일반 및 개선 알고리즘의 성능을 비교하여 나타내었다. 직렬 프로세서에서의 성능과 비교하여 볼 때 병렬 컴퓨터에서 최소 512배의 향상을 보였다.

**표 2.** 병렬컴퓨터에서 일반 평행사변형 알고리즘과 개선 평행사변형 알고리즘의 성능 비교

(단위: loop count)

분산값	일반 평행사변형 알고리즘	개선 평행사변형 알고리즘
	1	60928
2	40600	27958
3	27769	10675

**표 3.** 직렬 및 병렬 컴퓨터에서 일반 및 개선 알고리즘의 성능 비교

(단위: speedup)

	직렬		병렬	
	일반	개선	일반	개선
1 분산값	100	96.6	0.24	0.22
2 분산값	100	65.6	0.31	0.21
3 분산값	100	41.4	0.30	0.12

병렬 알고리즘의 성능은 speedup으로 나타낼 수 있다. speedup은 직렬 알고리즘의 수행시간을 병렬 알고리즘의 수행시간으로 나눈 값이다. 표 2의 결과로부터 일반 알고리즘과 개선 알고리즘의 speedup을 구한 결과가 표 4에 나타나 있다.

**표 4.** 병렬 알고리즘의 성능 비교

(단위: speedup)

	일반	개선
1 분산값	416.7	459.5
2 분산값	322.6	476.2
3 분산값	333.3	833.3

## 결 론

본 논문에서는 주어진 컴퓨터 모델에서 위성영상에 평행사변형 분류기법을 적용함에 있어서, 화소가 속한 군집의 초기값 설정이 알고

리즘의 성능에 미치는 영향을 분석하였다. 컴퓨터 모델은 직렬컴퓨터 모델과 병렬컴퓨터 모델중의 하나인 MIMD 모델을 대상으로 하였다. 본 연구에서 알고리즘의 성능을 개선하기 위해 제안하는 기본 발상은 위성영상내의 인접화소 간에 존재하는 공간적 상관성을 이용하는 것이다. 즉 임의의 화소의 군집을 정하는 단계에서 이미 결정된 이전 화소의 군집을 초기값으로 설정함으로써 탐색 'for' loop을 조기에 벗어나도록 한다.

먼저 직렬 컴퓨터 모델에서 화소의 분류율이 높은 2배 분산값, 3배 분산값을 문턱 값으로 삼은 경우 개선 알고리즘은 탐색 'for' loop의 실행회수에서 1.5배와 2.4배의 성능 향상을 보였다. MIMD 병렬 컴퓨터 모델에 따른 성능 변화를 비교하면, 전산모의 실험결과 512개의 프로세서를 사용한 경우 직렬컴퓨터에 비교할 때, 최소 512배의 성능 향상을 보였다. 그러나, 이 결과는 loop count를 기준으로 하였을 때의 결과이고, 실제 병렬컴퓨터에 수행시켜 실제 실행시간을 비교할 경우 프로세서간의 통신에 의한 시간 등과 같은 요인들 때문에 성능향상은 그보다 적을 것으로 예상된다. 실제로, 직렬 컴퓨터와 동일한 성능을 가진  $n$ 개의 프로세서를 사용하는 경우 성능은  $n$ 배를 넘을 수 없다. 직렬 컴퓨터에 알고리즘을 수행시킨 경우에도 단순히 덧셈과 곱셈에 필요한 명령문들의 실행 횟수만을 고려하여 성능향상을 측정하였으나, 실행시간을 비교하면 알고리즘의 변화에 따른 코드의 변화로 인하여 성능 향상 정도는 줄어들 것으로 예상된다.

본 연구를 통하여 위성영상의 감독분류기법인 평행사변형 알고리즘의 경우 군집의 초기값을 적절하게 설정하는 것이 알고리즘의 성능에 상당한 영향을 미치며, 직렬컴퓨터와 MIMD 병렬컴퓨터에서 초기값의 적절한 설정을 통해 분류기법의 성능이 향상됨을 확인하였다. **KAGIS**

## 참고문헌

- 정경훈. 2000. 다중스펙트럼 위성영상 압축을 위한 복합부호화 기법. 한국지리정보학회지 3(1):1-11.
- Almasi, G. S. and A. Gottlieb. 1994. Highly Parallel Computing. The Benjamin/Cummings Publishing Co, 2nd.
- Bryant, J. 1979. On the clustering of multidimensional pictorial information. Pattern Recognition 11:115-125.
- Bryant, J. 1989. A fast classifier for image data. Pattern Recognition 22(1):45-48.
- Hwang, K. and F. A. Briggs. 1984. Computer Architecture and Parallel Processing. McGraw-Hill Book Co.
- Jensen, J. R. 1996. Introductory Digital Image Processing: A Remote Sensing Perspective. Prentice Hall, 2nd. **KAGIS**