

# 스캔라인을 이용한 LIDAR 포인트 cloud의 분리에 기반한 분리된 개체간 인접관계의 정립에 관한 연구

A study on settlement of neighboring relation of  
segmented objects based on segmentation of LIDAR point  
cloud utilizing scan line characteristics

한수희\*, 이정호\*, 유기윤\*\*, 김용일\*\*\*

(Soohee Han, Jeongho Lee, Kiyun Yu, Yongil Kim)

서울대학교 지구환경시스템공학부 \*박사과정, \*\*조교수, \*\*\*부교수

## 초록

본 연구에서는 스캔라인을 이용한 LIDAR 포인트 cloud의 분리과정 중 분리된 포인트 군집간 인접 관계를 인식할 수 있는 기능을 추가하였다. 군집간 인접관계는, 포인트 cloud 분리 과정 중에 분리된 건물 요소를 재결합하거나 지면 포인트를 인식하기 위하여 사용될 수 있다. 실험 결과 포인트 cloud 분리 과정에 군집간 인접 관계 인식 기능을 추가하더라도 처리 성능이 저하되지 않았으며 후처리를 통하여 건물 요소를 결합하여 온전한 형태의 건물 포인트 군집을 형성함과 더불어 지면 포인트 군집도 인식할 수 있음을 확인하였다.

## 서론

항공 레이저 측량 시스템의 스캔라인 특성을 이용한 효율적인 포인트 cloud의 분리([1], [2])에서는 원시 LiDAR 포인트 자료를 분리하기 위하여 region growing 과 무감독 분류 방식을 응용하였으며 스캔라인의 특성을 이용하여 처리 성능을 개선하기 위한 방법론을 제시하였다. 제시된 알고리즘에 의해 분리된 각 포인트 군집은 다른 군집과 일정 거리 이상 떨어져 있거나 고도가 주어진 값 이상으로 차이가 나게 된다. 따라서 동일 건물에 속한

건물 요소들 역시 서로 고도 차이가 나면 독립적인 군집으로 분리된다. 따라서 포인트 cloud로부터 분리된 개체 군집 수는 실제로 존재하는 개체에 비해 매우 많은 수를 나타낸다. 본 연구에서는 이와 같은 건물 요소를 인식하기 위하여 군집간 인접 관계를 확인할 수 있는 기능을 기제시한 분리 과정에 추가하였다. 아울러 군집간 인접 관계는 지면 포인트 군집의 인식에도 활용될 수 있음을 확인하였다. 본 논문은 기제시된 포인트 cloud 분리 알고리즘의 보완된 형태와 후처리 과정, 적용 및 결론의 순서로 구성되어 있다.

## 본론

### 1. 기 제시된 포인트 cloud 분리 알고리즘

기 제시된 포인트 cloud의 분리 알고리즘 ([1], [2])에서 스캔라인을 따라 입력되는 신규 포인트는 기본적으로 새로이 생성될 군집의 seed 포인트가 되며, 기 생성된 군집에 신규 포인트와 고도 차이가 주어진 임계치 이하인 인접 포인트가 존재할 경우 해당 군집으로 분류된다. 만약, 신규 포인트가 두 개 이상의 군집으로 분류 가능할 경우 해당 군집들은 하나의 개체로 인식되어 통합된다(그림 1).

```

step 1
initialize n=1
step 2
initialize  $L_{merge}$  as empty and  $nc=i=0$ 
step 3
set  $d_{mi}=d(p_{new}, p_{mi}), h_{mi}=h(p_{new}, p_{mi})$ 
step 4
if  $(d_{mi} \leq t_{dist} \& h_{mi} \leq t_{height})$  then
 $nc=nc+1$ 
go to step 3 for  $\forall p_{mi} \in G_i$ 
step 5
if  $(nc==1)$  then classify  $p_{new}$  into  $G_i$ 
and put  $i$  into  $L_{merge}$ 
else if  $(nc>1)$  then put  $i$  into  $L_{merge}$ 
set  $nc=0$  and go to step 3 for
 $G_i(i=1, \dots, n)$ 
step 6
if  $(n(L_{merge})==0)$  then  $n=n+1$ , create
 $G_n$  and classify  $p_{new}$  into  $G_n$ 
else if  $(n(L_{merge})>1)$  then merge
groups listed in  $L_{merge}$ 
repeat from step 2 to step 6 for
 $\forall p_{new} \in \{point\ cloud\}$ 

```

그림 1 포인트 cloud 분리 알고리즘

여기서,  $n$ : 총 군집 수,  $p_{new}$ : 신규포인트,

$p_{mi}$ :  $i$ 군집의  $m$ 번째 포인트,  $G_i$ :  $i$ 번째 군집,  $d(p_1, p_2)$ 와  $h(p_1, p_2)$ : 포인트  $p_1, p_2$ 의 거리와 높이차,  $L_{merge}$ : 통합될 군집 목록

### 2. 보완된 포인트 cloud 분리 알고리즘

군집간 인접 관계를 파악하기 위하여 다음과 같은 조건을 추가하였다. 신규 포인트가 기존 군집의 한 개 이상의 포인트와 인접하나 고도 차이가 임계치 이하인 포인트가 해당 군집에 존재하지 않는 경우, 신규 포인트를 seed 포인트로 갖는 군집은 해당 군집과 인접하는 것으로 인식된다. 따라서 (그림 1)의 "step 4"는 (그림 2)와 같이 수정된다.

```

step 4
if  $(d_{mi} \leq t_{dist} \& h_{mi} \leq t_{height})$  then
 $nc=nc+1$ 
else if  $(d_{mi} \leq t_{dist} \& h_{mi} > t_{height})$  then
put  $i$  into  $N_{temp}$ 
go to step 3 for  $\forall p_{mi} \in G_i$ 

```

그림 2 수정된 step 4

여기서,  $N_{temp}$ : (임시)인접 군집 목록

다음으로 (그림 3)과 같이 "step 7"이 추가되어 신규 포인트가 최종적으로 분류된 군집에 인접 군집 목록을 첨부하며 인접 군집쌍은 서로에 대한 인접 관계 정보를 추가한다.

```

step 7
copy  $N_{temp}$  to  $N_k(p_{new} \in G_k)$  and put
 $k$  into  $N_j(j \in L_k)$ 
empty out  $N_t$ 

```

그림 3 추가된 step 7

군집간 인접 관계 파악 방식을 도식화 하면 (그림 4)와 같다. (그림 4)에서 스캔라인을 따라 포인트 p\_A가 입력되면 p\_A는 group A로 분류될 수 있다. 다음으로 입력되는 포인트 p\_B는 group A의 포인트들과 일정 수준 이상의 고도 차이가 나므로 group A로 분류될 수 없다. 따라서 p\_B는 새로운 group B의 seed point가 된다. 그러나 p\_B는 p\_A와 인접해 있으므로 group B와 group A는 서로 인접해 있는 것으로 인식되며 group A와 group B에 서로에 대한 인접 정보가 추가된다.

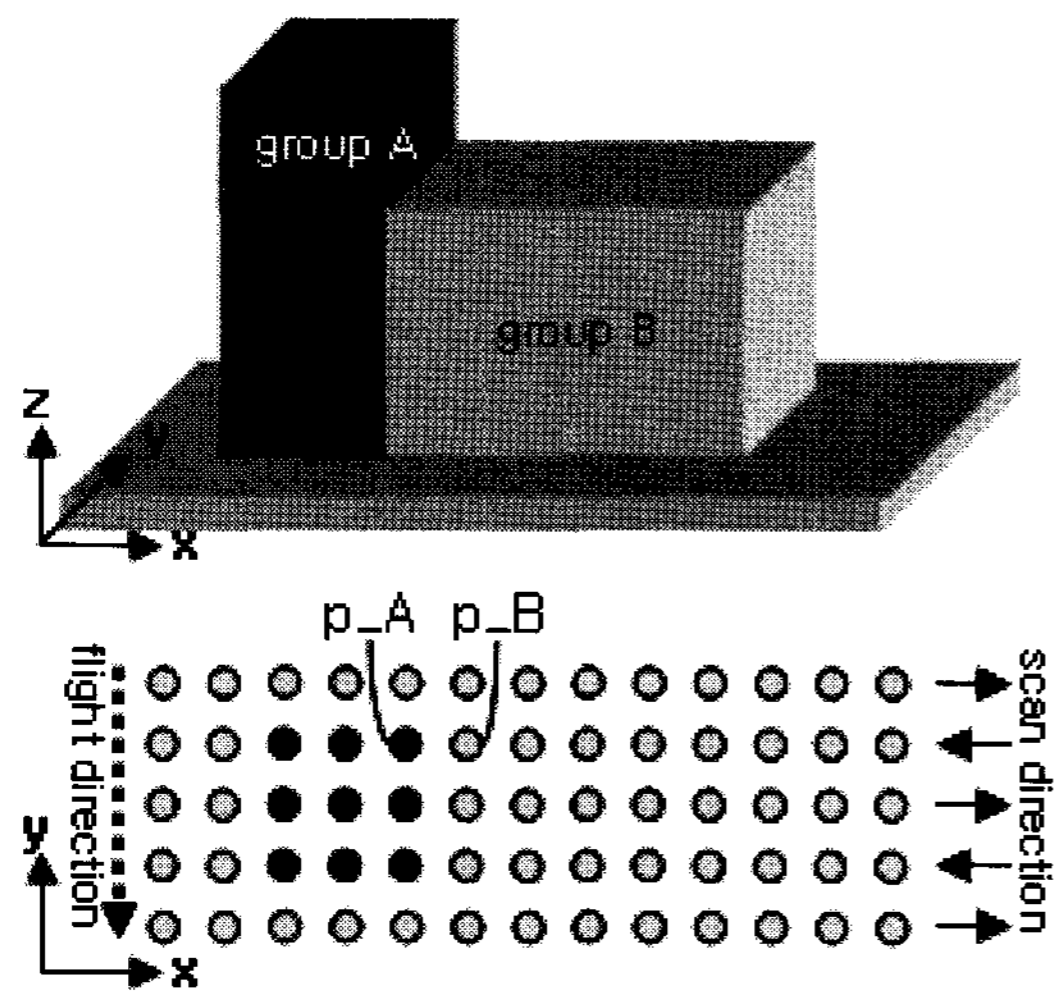


그림 4 개체간 인접 관계의 파악

한편, "step 6"에서와 같이 어떤 군집이 다른 군집으로 통합될 경우 인접 정보 역시 통합되어야 하며 통합된 군집과 인접한 군집들의 인접 정보를 갱신해주어야 한다.

### 3. 군집간 인접 정보를 이용한 후처리

군집간 인접 정보를 이용한 분리후처리 과정은 다음과 같이 구성된다.

- 지면 포인트 군집 인식

- 동일 건물의 건물요소 결합 및 재 레이블링
- 삭제 개체를 고려한 인접 관계 재정립

#### 가. 지면 포인트 군집 인식

기 제시한 포인트 cloud 분리 방식([1], [2])은 지면을 지면으로 인식하는 것이 아니라 건물 등과 마찬가지로 다른 하나의 군집으로 분리하는 것이다. 즉, 결과적으로 지면 군집이 분리되었으나 어느 군집이 지면을 나타내는지는 명시되지 않은 상태로 종결된다. 현재까지 지면 포인트를 인식하기 위한 다수의 필터링 방식이 개발되었으나 본 연구에서는 군집간 인접 정보만으로 간단히 지면 군집을 인식할 수 있는 방법을 제시하였다.

일반적으로 지상의 개체들은 일부 건물요소를 제외하고 대부분 지면과 접해 있다. 따라서 대상 지역이 충분히 크고 지면이 축대나 기타 인공구조물로 인하여 여러 부분으로 나뉘어 있지 않는 한 지면 군집은 상대적으로 매우 많은 수의 인접 군집을 갖게 되며 이와 같은 특성을 이용하면 지면 군집을 인식할 수 있다. 실제로 300여만 개의 샘플 포인트를 처리하였을 경우 (표 1)과 같은 결과를 나타내었다.

총 군집 수	44,198개
인접 군집 수 평균	약 5개
인접 군집 수 중앙값	3개
인접 군집 수 표준편차	약 125
지면 군집의 인접 군집 수	26,259개

표 1 지면 군집의 인접 군집 수

#### 나. 건물요소 결합 및 재 레이블링

서로 다른 고도로 구성된 건물 요소는 인접 관계를 지니게 되므로 이를 이용하여

분리된 이후에 재결합 할 수 있다. 즉 (그림 5, 6)에서와 같이 인접 관계에 있는 군집에 동일한 주 ID를 부여하고 상세 모델링을 위한 보조 ID를 부여한다.

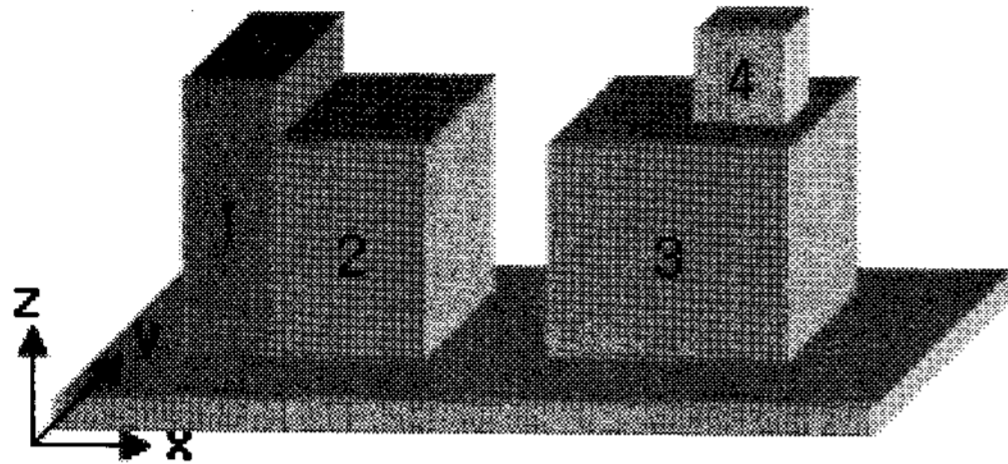


그림 5 분리된 건물 요소

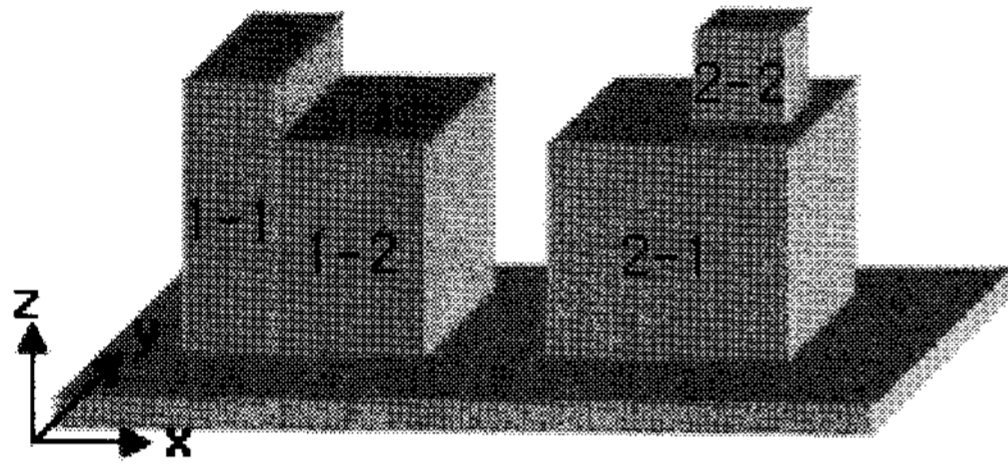


그림 6 재결합된 건물 요소와 재레이블링

다. 삭제 군집을 고려한 인접 관계 재정립

독립된 건물들이 오차 포인트나 수목, 인접한 불필요 군집 등에 의하여 건물 요소로 인식되는 경우 이들을 제거한 후에 군집간 인접 관계를 재정립 한다(그림 7, 8). 그러나 수목 등 제거 대상 인식 방법에 대해서는 본 논문에 언급하지 않았다.

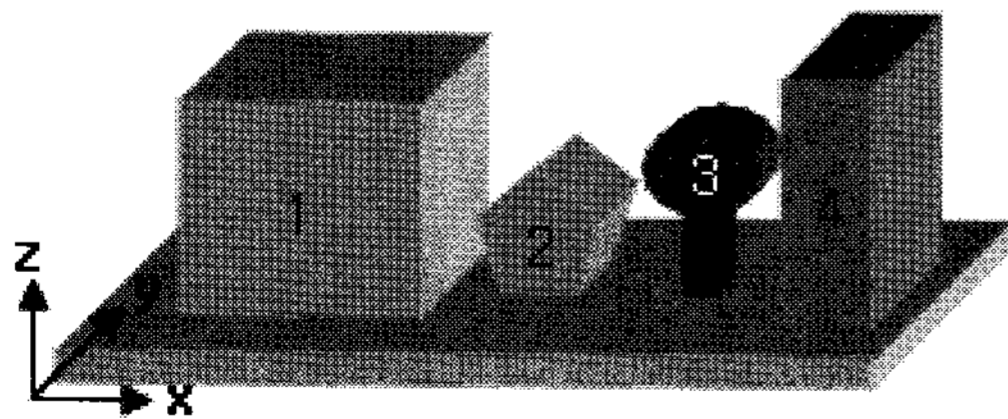


그림 7 건물, 불필요 군집, 수목 등이 인접한 경우

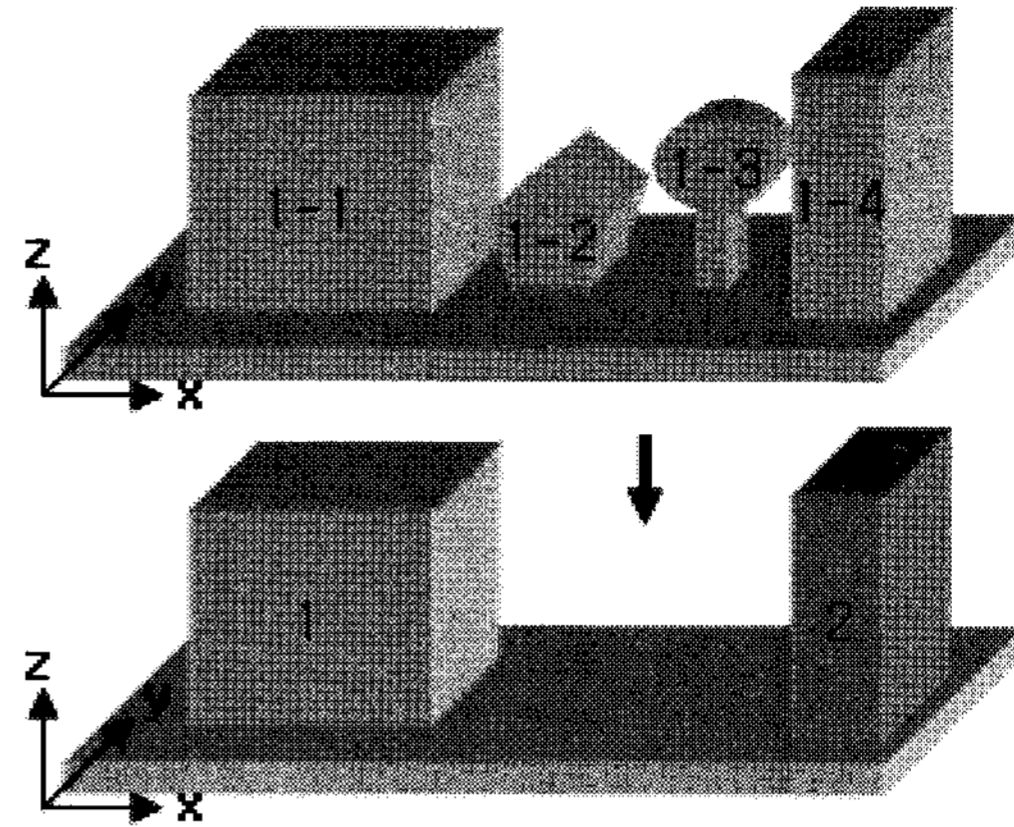


그림 8 군집 삭제와 인접관계 재정립

#### 4. 적용

기 제시한 포인트 cloud의 분리 알고리즘 ([1], [2])과 군집간 인접 관계 파악을 위해 보완된 알고리즘의 성능을 평가하기 위하여 처리 시간과 순간 최대 메모리 사용량을 비교하였다. 결과적으로 처리 성능에는 거의 차이가 없음을 확인할 수 있었다(표 2). 사용된 샘플 자료와 처리 시스템의 제원은 (표 3)과 같다.

	처리 시간	메모리 사용
수정 전	2분 10초 내외	약 122MB
수정 후	2분 10초 내외	약 122MB

표 2 알고리즘의 처리 성능 비교

ALS 시스템	
모델명	ALTM 3070
제작사	Optech Inc.
총 포인트 수	3,066,991개
점밀도	약 1.5point/m <sup>2</sup>
처리 시스템	
CPU	AMD Athlon XP 2800+
RAM	1 GB
언어	Visual C++ 6.0

표 3 샘플 자료와 처리 시스템의 제원

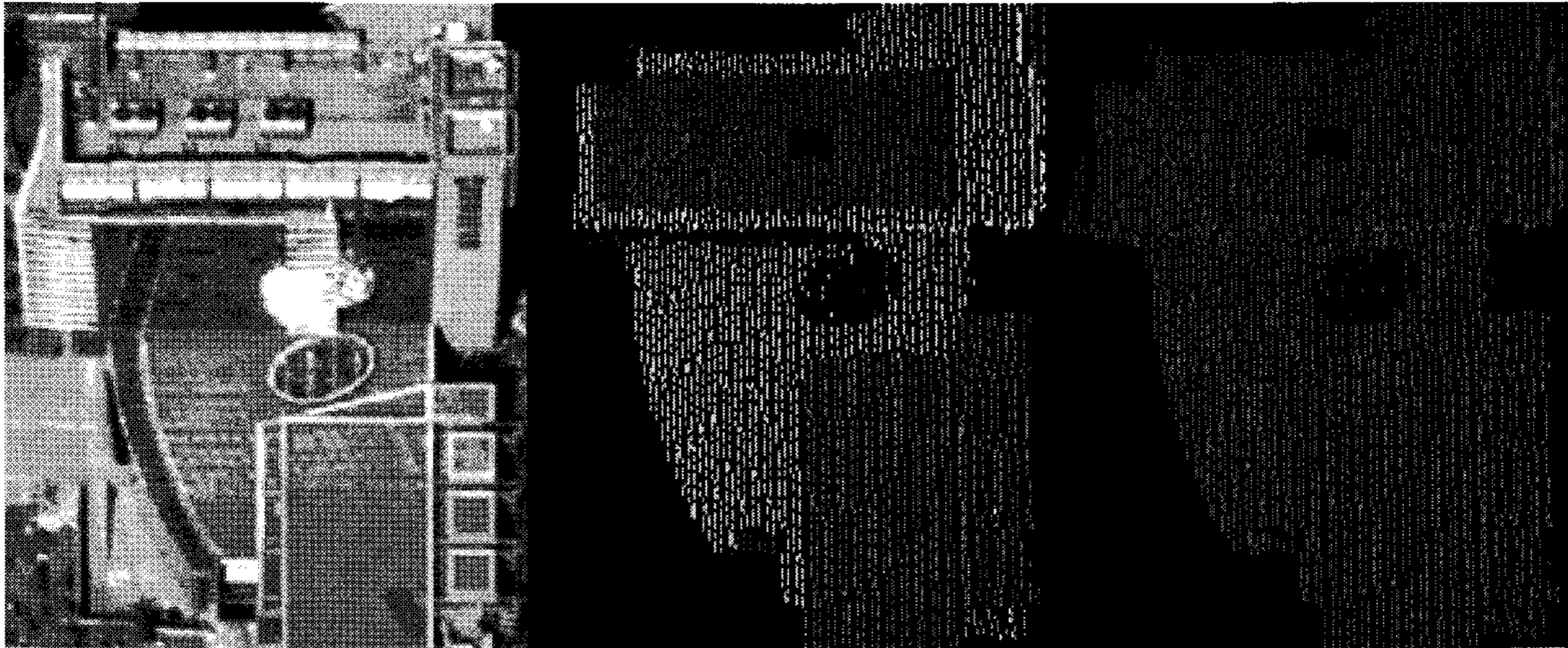


그림 9 레퍼런스 영상(좌), 건물 요소의 인식(중), 건물 요소의 결합(우)

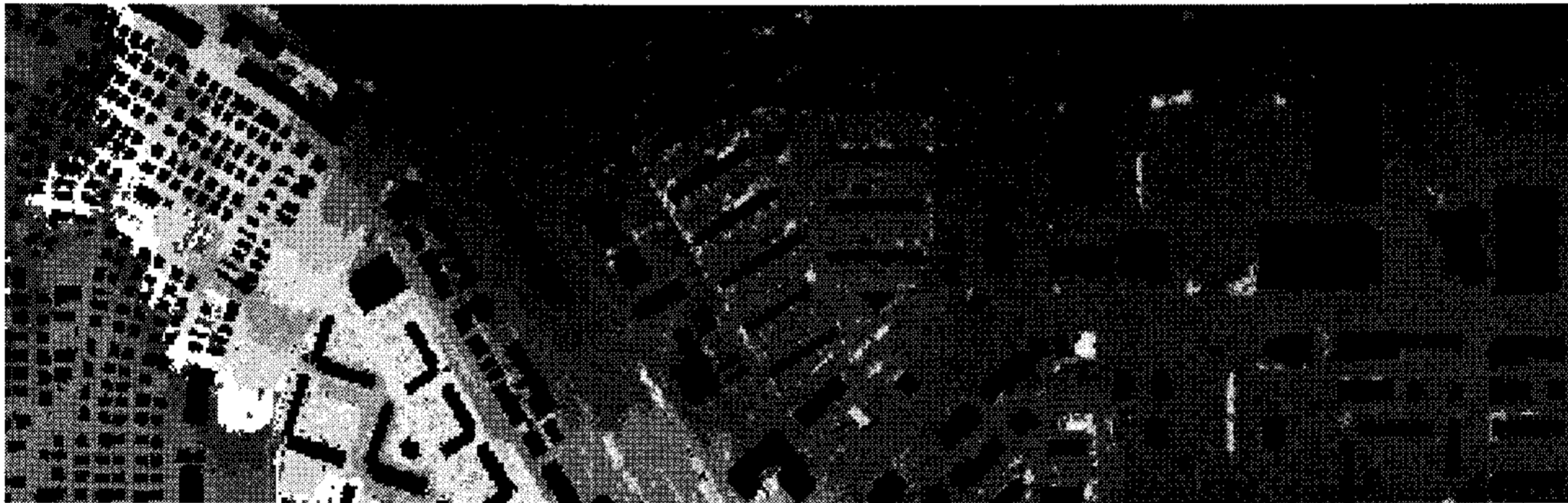


그림 10 지면 포인트

건물 요소로 분리되었던 아파트의 옥탑과 복잡한 형태의 지붕으로 구성된 건물의 건물 요소들이 대부분 인식되었으며 이들을 결합하여 온전한 형태의 건물을 표현할 수 있었다(그림 9).

또한 지면 포인트 군집도 인접 군집 수가 평균 인접 군집 수의 3 sigma를 넘는다는 조건을 통하여 인식함으로써 그 활용 가능성을 제시하였다(그림 10). 그러나 인공 구조물로 인해 고립되어 비교적 적은 수의 인접 관계를 갖는 소규모 지면 군집은 제시한 방식으로는 인식할 수 없다는 한계점도 확인할 수 있었다.

## 5. 결론

본 연구에서는 스캔라인을 이용한 LIDAR 포인트 cloud의 분리과정에 분리된 포인트 군집간 인접 관계를 인식할 수 있는 기능을 추가하였다. 적용 결과 기능의 추가로 인한 분리 과정의 처리 성능 저하는 거의 발생하지 않았다. 분리된 군집간 인접 관계를 이용하여 건물 요소를 인식하고 재결합할 수 있었으며 지면의 인식에도 활용할 수 있음을 확인하였다.

따라서 방대한 분량의 LiDAR 포인트 자료의 효율적이고 효과적인 처리는 물론 분리된 건물 요소의 인식과 결합을 통한

온전한 형태의 건물 복원, 건물 요소의 개별 모델링을 통한 상세 모델링에 기여할 수 있을 것으로 보인다.

#### 참고문헌

[1] 한수희, 이정호, 유기윤, 2005, 항공 레이저측량 자료의 스캔라인 특성을 활용한 건물 포인트 분리에 관한 연구, 한국지형공간정보학회논문집, 제 13권, 4호, pp. 33~38

[2] 한수희, 이정호, 유기윤, 2006, ALS의 스캔라인 특성을 이용한 효율적인 포인트 클라우드의 분리, 공동춘계학술대회, 2006