

객체인식과 분석을 위한 3D 체인코드의 적용

Application of 3D Chain Code for Object Recognition and Analysis

박소영¹⁾ · 이동천²⁾

Park, So Young · Lee, Dong-Cheon

Abstract

There are various factors for determining object shape, such as size, slope and its direction, curvature, length, surface, angles between lines or planes, distribution of the model key points, and so on. Most of the object description and recognition methods are for the 2D space not for the 3D object space where the objects actually exist. In this study, 3D chain code operator, which is basically extension of 2D chain code, was proposed for object description and analysis in 3D space. Results show that the sequence of the 3D chain codes could be basis of a top-down approach for object recognition and modeling. In addition, the proposed method could be applicable to segment point cloud data such as LiDAR data.

Keywords : Object recognition, 3D chain code, Data segmentation, Shape analysis, Modeling

초 록

객체의 형상을 결정할 수 있는 요소는 크기, 경사도와 경사방향, 곡률, 기복변화, 요소간의 특성관계, 선 또는 면이 이루는 각, 특징점의 분포형태 등 매우 다양하다. 객체가 존재하는 공간은 3차원이지만 대부분의 경우 2차원 공간상에서 객체를 표현하고 이를 기반으로 객체인식을 수행하고 있다. 본 연구에서는 3차원 공간에서 객체의 형태를 판단하기 위한 방법을 제안하기 위하여 기존의 체인코드를 3차원으로 확장시켜 큐브형태의 연산자의 적용을 수행하였다. 일련의 코드로 생성되는 3D 체인코드를 분석하여 객체를 정형화하여 하향식 방법에 의한 객체 모델링의 기반이 될 수 있음을 보여 주었다. 또한 점 데이터 분할에 적용할 수 있으며 이는 라이다 데이터에 의한 모델링에 사용될 수 있다.

핵심어 : 객체인식, 3D 체인코드, 데이터 분할, 형태분석, 모델링

1. 서 론

공간정보의 주요 연구분야는 3차원 공간상의 지형·지물에 대해 획득한 데이터로부터 객체를 인식하고 특성을 파악하여 필요한 정보를 추출하는 것이다. 약 30년 전부터 여러 분야에서 객체인식 방법이 연구되고 있으며, 특히 영상센서에 의한 디지털 영상매체가 이용되기 시작하면서, 수치영상처리 기법과 컴퓨터비전 패러다임이 본격적으로 정립되기 시작하였다(Marr, 1982). 수치사진측량과 수치지도 분야의 발전을 위해 필요한 기술은 high-level vision processing 기반의 객체인식이며, 이를 이용한 데이

터 처리 자동화와 신뢰성 향상을 위한 연구가 진행되어야 한다(Blake and Troscianko, 1990; Schenk, 1999).

대부분의 객체인식 방법은 2차원 영상을 기반으로 하고 있다. 그러나 촬영위치와 방향, 센서모델 등 대상물과 영상간의 투영 조건에 따른 기하학적 왜곡에 의해 객체인식과 분석 결과가 다를 수 있다. 그러므로 2차원 영상에 투영된 대상물로부터 3차원 공간상에 존재하는 객체의 형상과 특성을 정확하게 인식하는 것은 한계가 있다. 또한 객체가 이동 또는 회전하거나 축척이 변하여도 객체인식은 일관성과 불변성이 유지되어야 한다. 즉, 그림 1 처럼 기하학적 변화(이동, 회전, 축척 등)와 노이즈에 영향을 받

1) 세종대학교 지구정보공학과 석박사통합과정(E-mail:sysypark@sju.ac.kr)

2) 교신저자·정회원·세종대학교 지구정보공학과 교수(E-mail:dclee@sejong.ac.kr)

지 않아야 하는 조건이 만족되어야 한다(Lehar, 2002).



그림 1. 기하학적 변화에 불변한 객체인식

영상으로부터 객체를 묘사하고 특징을 파악하는 방법은 주로 객체의 윤곽선을 추출하고 윤곽선의 기하학적 특성인 길이, 방향, 곡률, 특징점, 교차선의 형태 및 구성 등을 분석하는 체인코드, ψ -s 곡선, line labelling 방법 등이 있다(Sonka 등, 1999; Tanimoto, 1987). 그러나 2차원 데이터를 이용하여 객체를 인식하는 것은 한계가 있으므로 객체가 존재하는 3차원 공간에 적용할 수 있는 방법의 개발이 중요하다. 3D 체인코드에 대한 연구는 비교적 최근에 수행되고 있으며, 대표적인 연구로 Bribiesca(2000, 2004)는 두 선이 교차하는 상대적인 방향에 대하여 코드를 정의하고 이동과 회전에 불변한 체인코드로 3차원 곡선을 묘사하는 방법을 제안하였으며, 수치표고모델(DEM)의 등고선을 3D 체인코드로 변환하고 이를 이용하여 지형을 복원하였다. 한창호 등(2010)은 권투선수의 동작을 샘플링하고 정규화하여 큐브형태로 정의한 코드와 3차원 좌표상의 각도를 수평과 수직면에 투영한 성분을 이용하여 권투선수의 동작을 수치적으로 분석하였다. 본 연구에서 추구하는 라이더 데이터에 의한 객체 모델링의 대표적 연구로는 김성준 등 (2010)과 송낙현 등(2007)이 있으며, 좌표의 기하학적 특성과 관계를 분석하여 객체면 요소를 분할하고 추출하는 방법이 있다. 본 연구의 근간이 되는 2D 체인코드는 객체의 방향성 또는 경사에 따라 정의한 수치로부터 객체를 표현하는 방법으로써 이를 3차원으로 확

장하면 3차원 공간상에 좌표로 구성된 객체를 코드화하고 규칙성을 부여하여 정형화된 모델 데이터베이스를 구축할 수 있다. 그러므로 3D 체인코드로 변환된 객체를 정형화된 모델 데이터베이스와 유사성을 비교하여 하향적 방법(top-down approach)으로 객체의 특성을 파악하고 이를 기반으로 모델링이 가능하다. 또한 코드를 분석하여 기하학적 특성에 따른 데이터 분할에 활용할 수 있다(박소영 등, 2011).

본 연구에서는 3차원으로 확장된 체인코드를 이용하여 수치지형도와 같은 3차원 벡터 데이터와 라이더 데이터와 같이 3차원 점의 좌표로 구성된 객체를 코드화하여 묘사하는 방법을 제안하였으며, 제안한 방법을 구현하기 위하여 객체의 상부모양에 따라 평면형, 계단형 및 박공형을 실험 데이터로 사용하였다. 또한 라이더 데이터의 객체 분할에 사용 가능성을 파악하기 위하여 피라미드형과 박공형으로 구성된 복합 지붕형태의 시뮬레이션 데이터와 실제 LiDAR 데이터에 적용하여 결과를 분석하였다.

2. 3D 체인코드

2.1 체인코드

체인코드의 기본 원리는 객체 윤곽선의 방향과 길이 성분을 규칙적인 수치로 표현하는 방법으로써 일련의 코드로 부호화하여 표현하고 회전, 축척 및 이동에 대한 불변성이 보장되어야 한다. 즉 기하학적 변화가 발생하여도 생성한 코드는 일정한 규칙성이 보존되어야 하며, 또한 역변환에 의하여 객체를 재현할 수 있어야 한다.

체인코드는 1961년에 Freeman에 의해 제안되었으며, Freeman chain code(FCC), Papert chain code(PCC), Chain code histogram(CCH), Chain code coherence vector(CCCV), Chain code distribution vector(CCDV), Vertex chain code(VCC), Extended vertex chain code(E_VCC), Variable length vertex chain code(V_VCC), Compressed vertex chain code(C_VCC), 3OT chain code(3OTCC) 등 여러 가지 개선된 방법들이 개발되었다(Wulandhari and Haron, 2008). 그러나 기존의 체인코드는 2차원 객체에 국한하여 4개 또는 8개의 평면 방향성만 고려하고 있으므로, 3차원 공간상의 객체를 2차원 평면으로 투영하여 객체모사와 형태분석에 사용하는 것은 문제점이 있다.

2.2 3D 체인코드의 특성

그림 2와 같이 $3 \times 3 \times 3$ 의 큐브 코드(한창호 등, 2010)를

이용하여 객체의 형태를 묘사한다. 이를 위하여 3차원 방향성을 판별하기 위하여 큐브의 중심을 기준으로 26개의 방향을 나타내는 코드로 정의하였다. 코드 생성을 위하여 큐브를 이동시키고 큐브 중심에 객체의 존재를 확인할 수 있는 데이터 값이 존재하면 체인코드를 생성하기 시작한다. 큐브 중심을 기준으로 객체의 윤곽선 또는 격자점 정보를 이용하여 방향성을 계산하여 이에 대응되는 코드값을 부여한다.

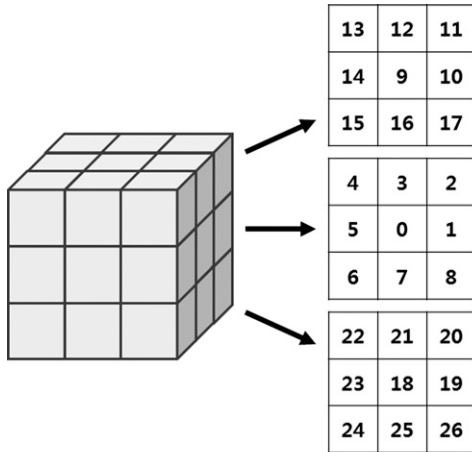


그림 2. 3D 체인코드의 구성

3. 체인코드의 적용

체인코드는 여러 분야에서 활용되고 있으며 응용하는 방법도 다양하다. 본 연구에서는 기존에 제안된 3D 체인코드를 객체의 형태인식과 모델링을 위한 데이터 분할에 적용하고 결과를 분석하여 활용가능성을 연구하였다.

3.1 Model key point 기반

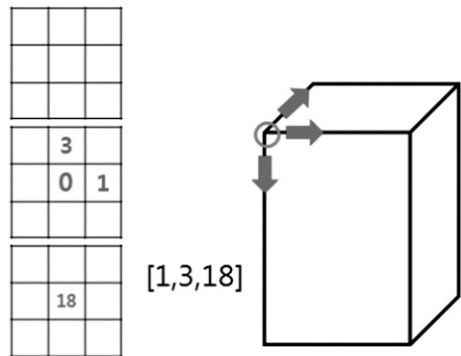
3.1.1 Model key point 추출

객체의 정확한 형태를 묘사하기 위해서는 객체의 특징을 나타낼 수 있는 충분한 개수의 데이터가 균형있게 존재하여야 한다. 즉, 최적의 데이터 구성이 필요하므로 객체를 구성하는 주요하고 필수적인 model key point를 추출하여야 한다. 3D 체인코드를 적용하여 객체의 윤곽선을 인식하여 형태를 분석할 수 있을 뿐만 아니라 객체를 구성하는 특징점인 model key point를 판별할 수 있다.

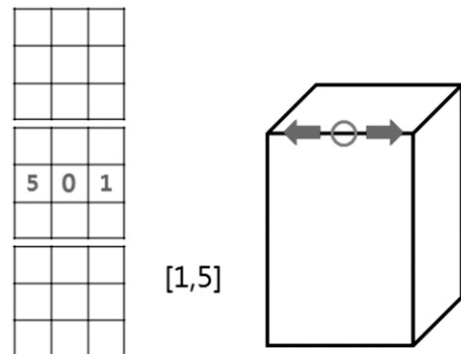
이선근 등(2007)은 지형적 특성을 최대한 유지하고 묘사할 수 있는 최적의 특징점들을 구성하기 위하여 등고선상의 vertex 간의 거리와 경사를 고려하여 model key point

를 추출하였다. Haralick 등(1992)은 영상에서 픽셀의 밝기 값이 급격하게 변하는 특징점(interest point)을 추출하여 객체의 특성분석과 영상정합에 이용하였다. 이와 같이 객체의 고유한 형태와 특성을 규정하고 묘사할 수 있는 model key point 모델링에 필수적인 공간정보를 포함하고 있다. 3D 체인코드로부터 model key point를 판별하기 위하여 큐브가 객체의 윤곽선을 따라 3차원 방향을 결정하여 체인코드를 부여한다.

코드의 개수가 3개 이상 부여되면, 그 지점에서 3개 이상의 방향성이 존재하는 모서리 점 (또는 교차점인 junction)이며 model key point라고 판단 할 수 있다. 예를 들면, 그림 3 (a)와 같이 큐브의 중심(0)에서 부여된 코드값이 [1,3,18]으로 3개 이므로 그 점은 model key point가 되며, 그림 3 (b)에서는 2개의 코드값인 [1,5]가 부여되어 객체의 윤곽선상의 한 점임을 알 수 있다. 또한 생성된 코드의 개수가 2개인 경우는 방향성이 일정한 직선임을 의미하므로 이 직선을 따라 큐브를 이동하면서 동일한 코드값이 부여된다.



(a)Model key point에 대한 체인코드



(b)윤곽선에 대한 체인코드

그림 3. 방향성 판단 및 체인코드 부여

3.1.2 3D 체인코드 생성

3D 체인코드는 model key point를 기반으로 큐브를 이동하면서 각 지점에서 대응하는 코드값을 부여함으로써 생성된다. 큐브가 이동하는 방향으로 계속해서 동일한 코드가 부여되면 그 코드값에 대응하는 방향으로 큐브를 이동하며 코드를 생성한다.

이와 같이 코드값에 부합되는 방향을 따라 이동하는 큐브에 더 이상 이전 위치에서 생성한 코드와 동일한 코드가 추출되지 않으면 한 번에 대한 체인코드 생성이 종료되었음을 의미한다. 또한 그 지점에서 다시 방향성에 따라 코드를 부여하고 방향을 나타내는 코드값이 3개 이상 존재하면, 그 점은 새로운 model key point가 되므로 다시 체인코드 생성을 시작한다. 이 때, 만일 3개의 코드 중 이전의 과정에서 이미 큐브가 이동하여 체인코드를 생성한 방향이라면 그 방향을 제외하고, 나머지 코드값 중 하나를 선택하고 큐브를 이동하여 위와 동일한 방법으로 코드를 생성한다. 선택할 코드값의 우선 순위는 수평방향 요소를 우선으로 하며, 수평요소에 대한 체인코드가 부여되

면 수직방향 요소에 의해 큐브를 이동시킨다. 그림 4와 그림 5는 3D 코드 생성 과정을 보여주고 있다.

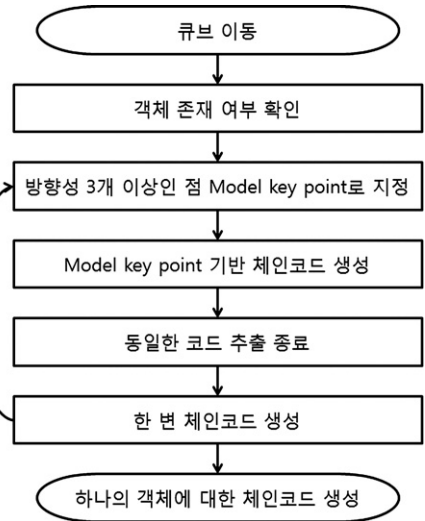


그림 4. 3D 체인코드 생성 과정

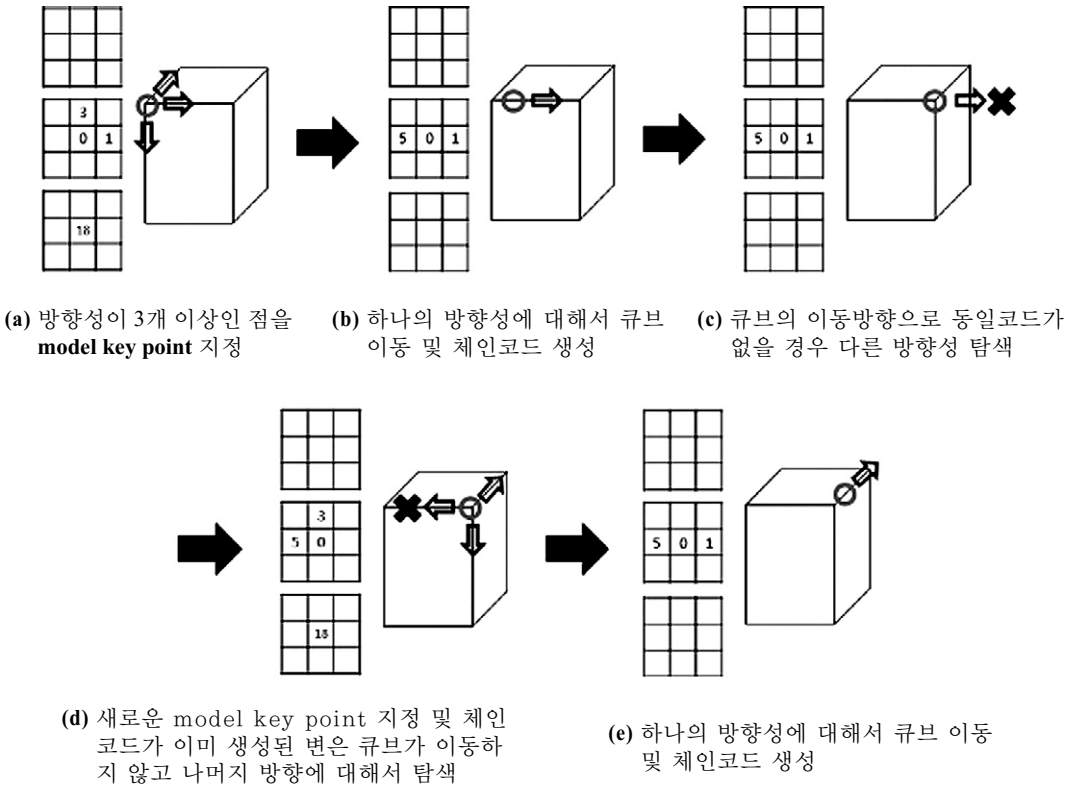


그림 5. 체인코드 생성 과정 모식도

이와 같은 과정을 반복하여 하나의 객체에 대해 체인코드를 생성하게 되며, 처음 시작점으로 돌아오게 된다. 이는 객체의 한 면에 대해 코드가 완성된 것이다. 그리고 새로운 면에 대한 체인코드를 생성하거나 수직요소의 연산자에 대해 생성한다. 결정해야 할 코드가 더 이상 존재하지 않으면, 한 객체에 대한 코드 생성이 종료되어 3차원 좌표로 이루어진 객체가 3D 체인코드로 변환되어 묘사된다. 또한 일련의 코드로부터 방향과 길이 성분도 연산되므로 객체의 축척 또는 크기 분석을 할 수 있다.

3.1.3 체인코드의 규칙성 분석

체인코드는 일련의 코드로서 객체묘사에 이용할 수 있지만, 생성된 코드가 절대적인 값이 아니므로 코드 자체만으로는 객체인식과 모델링에 이용하는 데 제약이 따른다. 그러므로 객체의 기하학적 변화에 의한 체인코드의 규칙성을 분석하여 객체인식에 필수적 요소인 불변성을 만족하는지 분석하여야 한다. 동일한 객체라도 회전된 경우 체인코드는 다르게 생성되므로, 모든 경우에 대해 무수한 객체 모델을 데이터베이스화 하여 하향식 모델링에 활용할 수 없다.

그러므로 기하학적 변화에 영향을 받지 않고 객체의 동일성 여부를 확인하는 검사를 수행하여야 한다. 회전, 축척 및 이동과 같은 기하학적 변화는 객체간의 상대적인 변화이므로 코드값의 차이를 계산한 일련의 차분값 패턴에 대한 규칙성을 분석하여 객체간의 상호관계를 파악하여야 한다. 동일한 모양의 두 객체가 서로 회전관계가 있는 경우, 다음 두 가지 방법으로 규칙성을 파악할 수 있다.

- 각 객체별로 객체 내에서 인접 코드간의 차분을 계산하면, 각각의 객체에 대해 특정 규칙성을 갖는 패턴으로 나타난다.
- 각 객체에 대해서 생성한 체인코드간의 차분을 계산하면, 규칙성을 갖는 패턴이 추출된다.

규칙성을 갖는 체인코드의 차분값을 방향성과 경사에 대한 정보를 갖고 있는 원래의 코드와 융합하여 회전에 불변한 동일성을 인식하게 된다. 즉 체인코드만으로는 회전 요소에 의한 객체의 변화를 감지할 수 없으므로 동일한 또는 유사한 객체를 모델 베이스에서 찾기 어렵다. 그러나 차분값을 이용하여 객체의 기하학적 변화를 판단하여 동일성을 결정하여 객체와 가장 적합한 모델을 결정할 수 있다.

3.2 Point cloud 데이터 기반

3.2.1 데이터의 격자화

라이다 데이터처럼 불규칙하게 분포된 점 데이터에 3D 체인코드를 적용하기 위해서는 데이터의 격자화가 요구된다. 격자화 결과는 격자 간격과 보간법에 따라 서로 다르지만, 본 연구에서는 격자화 자체가 주요 연구대상이 아니므로, 격자간격은 평균 점밀도를 고려하여 결정하였으며 최근린 보간법을 적용하였다.

3.2.2 3D 체인코드 생성

벡터 데이터인 수치지도와 같은 경우에는 model key point를 기반으로 객체 윤곽선의 방향성에 따라 코드를 생성하였지만, 점 데이터의 경우에는 모든 격자점에서 큐브를 종·횡방향으로 이동하여 객체에 대한 방향을 판단하여 코드를 생성한다. 객체의 모양에 따라 경사방향은 다양하지만 체인코드로 경사를 표현할 수 있는 최소 각도는 제한적이므로 그림 6과 같이 경사 범위를 설정하였다. 체인코드가 나타낼 수 있는 대표적 경사를 중심으로 $\pm 22.5^\circ$ 의 범위로 설정하였다.

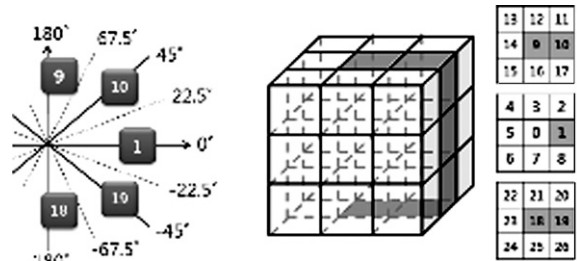
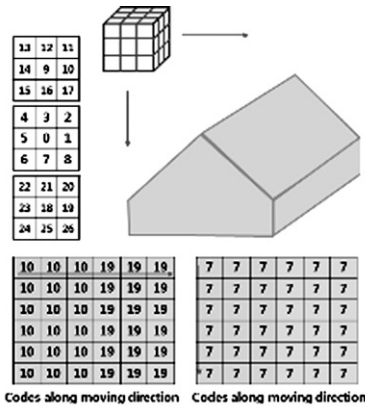


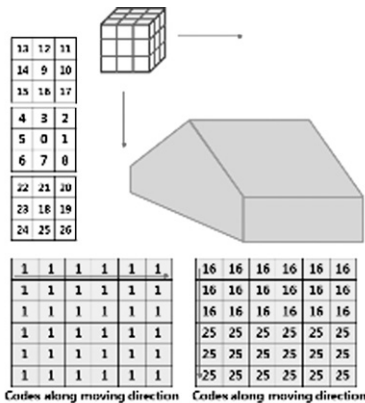
그림 6. 체인코드의 경사 범위

큐브가 이동하는 방향에 대하여 데이터 간의 경사를 결정하여 모든 격자점에서 코드를 부여한다. 객체의 회전에 의한 영향을 배제하고 객체의 경사방향에 관계없이 데이터를 분할하기 위하여 종방향과 횡방향으로 체인코드를 생성한다. 그림 7은 박공형 객체에 대한 체인코드 생성을 보여주고 있으며 큐브의 이동 방향에 따른 코드 생성 결과를 보여주고 있다. 그림 7(a)의 경우 좌우 방향에 대한 결과로부터 객체 분할이 가능하지만, 상하 방향에 대한 결과는 단일 코드로 추출되었다. 그림 7(b)의 경우는 그림 7(a)의 객체를 90도 회전한 것으로 (a)의 결과와 반대되는 결과가 획득된다. 이와 같이 큐브의 이동방향에 따라 적용한 결과를 분석하여 객체의 기하학적 특성을 파악할 수 있다. 이와 같은 결과를 기반으로 동일면에 존재하는 데

이터를 추출하여 객체를 구성하는 요소면을 분할(surface patch segmentation)할 수 있으며, 분할된 데이터를 이용하여 객체 모델링에 활용할 수 있다.



(a) 박공형 객체



(b) 객체의 90도 회전

그림 7. 격자형 데이터에 대한 체인코드 생성

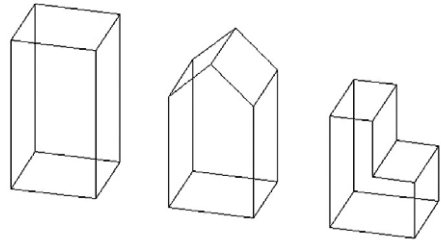
4. 실험결과 및 분석

4.1 실험 데이터

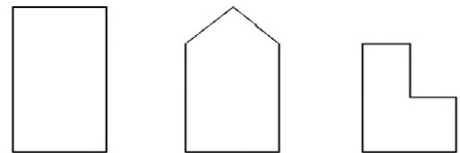
4.1.1 시뮬레이션 데이터

본 연구에서 제안한 방법의 적용과 결과 분석을 위하여 시뮬레이션 데이터를 구성하였다. Model key point 기반의 3D 체인코드 생성을 위한 시뮬레이션 데이터는 3차원 수치 도화된 데이터이며, 그림 8에서 보여주는 것처럼 객체의 상부 모양은 평면형(flat), 박공형(gable), 계단형(bi-level)이다. 박공형과 계단형의 평면도는 동일한 객체로 표현된다. 그러므로 2차원으로 투영하여 객체를 이용하면 다른

모양의 객체가 동일하게 인식될 수 있다. 이러한 문제를 해결하기 위해 3차원 공간상에서 적용되는 방법이 필요하다.



(a) 실험 객체의 3D 뷰



(b) 실험 객체의 정면도



(c) 실험 객체의 평면도

그림 8. 시뮬레이션 객체

그림 9는 Point cloud 데이터 기반의 3D 체인코드를 적용하기 위한 실험 데이터로서 항공 라이다 데이터를 시뮬레이션 하였으며, 점 간격은 25(단위없음)로 설정하였다. 무작위 노이즈는 X, Y, Z 좌표 각각 ± 10 (단위없음)로 부여하였고, 격자간격은 10(단위없음)으로 설정하였다. 점의 개수는 400×400 이다.

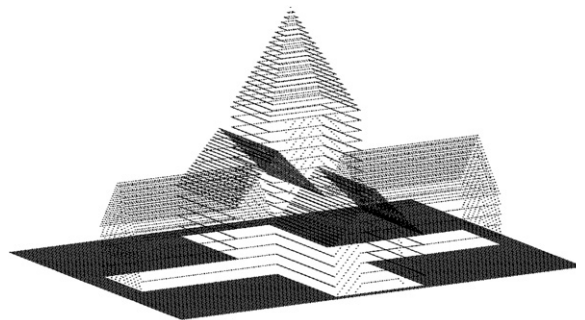
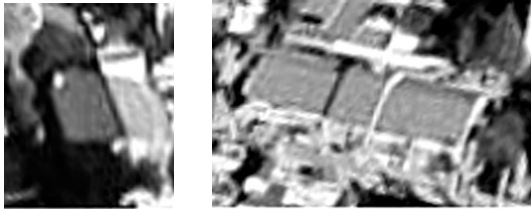


그림 9. Point cloud 시뮬레이션 데이터

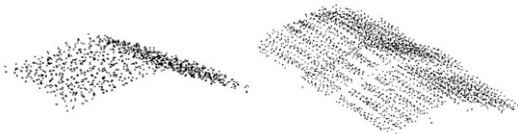
4.1.2 실제 LiDAR 데이터

실험에 사용한 LiDAR 데이터는 Optech의 ALTM 3070로부터 획득한 데이터이며, 점밀도는 6점/m²이다. 연구지역은 대전광역시 주택지역에서 박공형의 건물 2개를 선정하였다(그림 10 참조). LiDAR 데이터는 그림 11과 같으며, 건물 (a)는 단순 박공형, (b)는 중간의 높이가 다른 계단 박공형이다.



(a) 단순 박공형 (b) 계단 박공형

그림 10. 실험건물의 사진



(a) 단순 박공형 (b) 계단 박공형

그림 11. LiDAR 데이터

4.2 Model key point 기반

표 1은 3D 체인코드를 적용하여 추출된 객체의 model key point의 좌표값과 길이를 보여주고 있다. 표 2는 객체별로 3D 체인코드를 생성한 결과이며, 방향성 판단과 큐브 이동의 중복성을 확인하여 객체를 일련의 코드로 표현된 결과를 보여주고 있다. 공간상에서의 객체의 모양을 시각적으로 확인하여 형태나 특성을 파악하는 것이 아니라 새로운 도메인 또는 새로운 매개변수 공간에서 좌표정보가 코드값으로 변환됨을 알 수 있다.

그러므로 코드에 포함된 경사와 길이 요소를 기반으로 객체의 기하학적 형태를 추정할 수 있다. 또한 객체의 모든 좌표 데이터를 저장할 필요 없이 단순히 코드를 이용하여 객체를 인식하고 역변환에 의하여 객체를 재현할 수 있으므로 데이터 압축에도 활용할 수 있다고 판단된다. 또한 생성된 코드를 객체를 유형별로 정형화하고 그룹화하여 객체 모델 데이터베이스를 구축하여 하향적 방법으로 객체인식과 모델링의 기초 자료로 활용할 수 있다.

표 1. 실험 객체의 model key point 좌표 (단위 없음)

객체	X	Y	Z
평면형	60	20	40
	60	40	40
	80	40	40
	80	20	40
박공형	110	20	40
	110	40	40
	100	40	30
	100	20	30
	120	20	30
	120	40	30
계단형	20	20	30
	20	40	30
	30	40	30
	30	20	30
	30	40	15
	30	20	15
	40	20	15
	40	40	15

표 2. 실험 객체의 3D 체인코드

객체	체인코드	길이
평면형	1	20
	7	20
	5	20
	3	20
	18	40
박공형	1	20
	21	10
	5	20
	16	10
	25	10
	1	20
	12	10
계단형	18	30
	1	20
	7	10
	5	20
	3	10
	18	15
	5	20
	7	10
1	20	
3	10	
18	15	

즉 형태와 특성을 파악하고자 하는 특정 객체의 3D 체인 코드를 생성하고 이를 모델 데이터베이스와 유사성을 판별하여 모델링할 수 있는 기반이 될 수 있다.

표 3은 3D 체인코드는 차분에 의해 얻어지는 규칙성으로 객체의 동일성 여부와 객체의 회전에 불변함을 보여주고 있다. 시뮬레이션 데이터를 45도 회전시켜 코드를 생성하여 차분을 계산하였으며, 각 객체 내에서 인접한 코드간의 차분결과를 분석하여 유사한 패턴이 추출되었음을 알 수 있다. 객체가 회전된 경우 서로 다른 체인코드가 생성되므로 두 객체가 동일한 모양의 객체임에도 불구하고 동일한 형태의 객체로 인식할 수 없다. 그러나 차분에 의한 규칙성에 의하여 회전에 영향을 받지 않고 객체 모양의 동일성을 확인할 수 있다.

표 3. 실험 객체의 체인코드 차분값

객 체	회전 없음		45도 회전	
	체인코드	차분	체인코드	차분
평면형	1	-6	2	-6
	7		8	
	5	2	6	2
	3	2	4	2
	18	-15	18	-14
박공형	1	-20	2	-20
	21	16	22	16
	5	-11	6	-11
	16	-9	17	-9
	25	24	26	24
	1	-11	2	-11
	14		13	
	18	-6	18	-5
계단형	1	-6	2	-6
	7		8	
	5	2	6	2
	3	2	4	2
	18	-15	18	-14
	5	13	6	12
	7	-2	8	-2
	1	6	2	6
	3	-2	4	-2

표 4는 원래 객체와 회전한 객체 간의 코드를 차분한 결과이다. 두 객체가 회전에 관계없이 동일함을 확인하기 위하여 두 객체간의 코드값을 차분하였으며, 차분한 결과는 일정한 규칙성을 가지고 있음을 알 수 있다. 회전한 객

체의 코드는 원래 객체의 코드와의 차분결과 “-1” 값을 가지며, 이는 객체가 반시계방향으로 45도 회전한 것을 의미한다.

단, 표 3의 결과를 보면 각각의 체인코드 차분값이 유사하게 추출되었지만, 체인코드에서 수직 방향인 “18”의 경우(표 3에서 회색으로 표시한 부분), 회전에 영향을 받지 않기 때문에 동일한 체인코드가 생성되어 차분값이 서로 다르지만 그 차이가 일정하게 “1” 임을 주고 있다. 또한, 표 4에서도 코드값이 “18”인 경우(표 4에서 회색으로 표시한 부분)는, 회전에 영향을 받지 않으므로 코드가 변하지 않아 다른 차분값과 다르게 차분한 결과가 “0” 값을 갖는다. 그러나 이러한 결과는 규칙성에서 벗어나는 것이 아니라 이 또한 규칙성에 포함된다. 그러므로 차분값을 이용하여 객체간의 동일성과 기하학적 변화를 판단할 수 있다.

표 4. 회전에 의한 객체간 체인코드의 차분값

객 체	체인코드		차분값
	회전 없음	45도 회전	
평면형	1	2	-1
	7	8	-1
	5	6	-1
	3	4	-1
	18	18	0
박공형	1	2	-1
	21	22	-1
	5	6	-1
	16	17	-1
	25	26	-1
	1	2	-1
	12	13	-1
	18	18	0
계단형	1	2	-1
	7	8	-1
	5	6	-1
	3	4	-1
	18	18	0
	5	6	-1
	7	8	-1
	1	2	-1
	3	4	-1
		18	18

4.3 Point cloud 데이터 기반

4.3.1 시뮬레이션 데이터

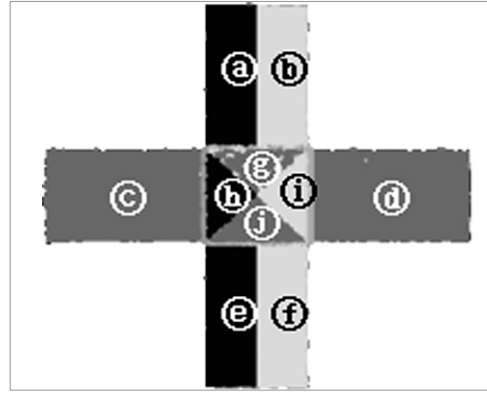
표 5는 격자화된 시뮬레이션 데이터에 큐브를 중·횡 방향으로 이동시켜 코드값을 부여한 것이며, 그림 11은 이를 기반으로 객체 요소면을 분할한 결과를 보여주고 있다. 객체에 대하여 큐브를 횡방향으로 이동한 결과 좌우 방향의 경사변화가 추출되었으며 (그림 12(a) 참조), 이에 해당하는 체인코드로부터 객체가 분할된 모습을 확인할 수 있다.

이와 같이 생성된 체인코드와 객체의 높이값 또는 기울기가 변하는 지점에서 이에 해당하는 코드를 이용하여 객체를 구성하는 단위 요소면을 분할할 수 있다. 즉 체인코드의 경사 또는 방향 성분으로부터 객체의 형태를 인식하고 묘사할 수 있을 뿐만 아니라 객체를 단위 요소면 별로 분할할 수 있다. 큐브의 종방향으로의 이동에 의한 결과 (그림 12(b) 참조)도 마찬가지로 객체의 형태파악과 단위 요소면으로 분할된 결과를 확인할 수 있다. 분할된 데이터를 이용하여 객체를 구성하는 단위 요소면에 대하여 최적의 수학적 함수를 이용하여 객체 모델링을 수행할 수 있다(김성준 등 2010; 이진형 등, 2010). 또한 중·횡방향으로 생성된 결과로부터 객체의 형태를 파악할 수 있다.

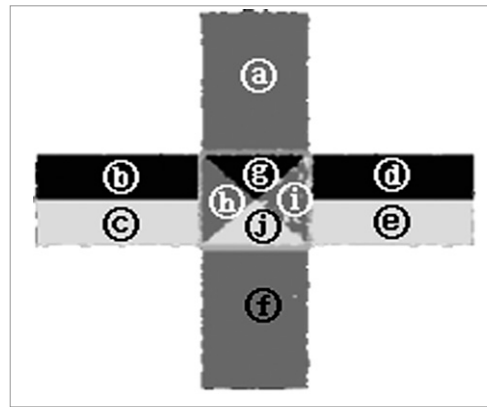
그러나 노이즈에 의해 실제 객체가 체인코드에 모 정확하게 표현 될 수 없는 경우도 발생한다. 그림 12의 (a)의 ㉔, (b)의 ㉑에는 각각 단위 요소면에 부합하지 않는 체인코드가 생성된다. 이 경우가 노이즈에 의해 발생한 오차이다. 이러한 오차는 전체 면적에 비해 작은 비율에 해당하기 때문에 데이터 분할시 무시할 수 있다.

표 5. 각 패치에 대한 체인코드

횡 방향		종 방향	
패치	체인코드	패치	체인코드
㉑	10	㉑	7
㉒	19	㉒	16
㉓	1	㉓	25
㉔	1	㉔	16
㉕	10	㉕	25
㉖	19	㉖	7
㉗	1	㉗	16
㉘	10	㉘	7
㉙	19	㉙	7
㉚	1	㉚	25



(a) 횡방향 적용



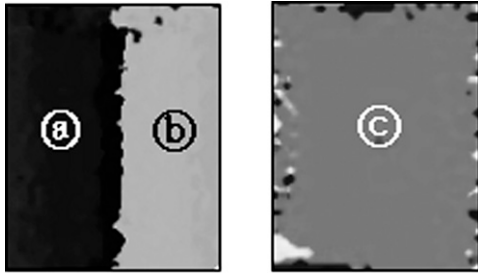
(b) 종방향 적용

그림 12. 체인코드에 의한 면분할 결과

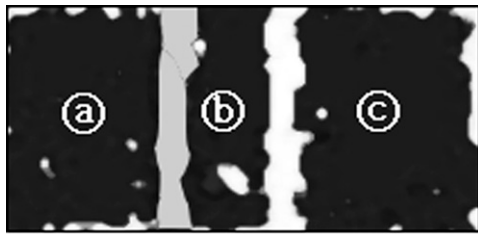
4.3.2 실제 LiDAR 데이터

실험 결과는 그림 13과 같다. 건물 (a)와 (b)에 대하여 큐브를 횡방향과 종방향으로 적용하여 박공형의 특성을 나타냄을 알 수 있었다. 건물 (a)의 종방향의 경우는 평면을 나타내는 코드가 생성되었으며, 건물 (b)의 종방향의 경우에는 지붕의 높이가 달라지는 중앙부분에서 높이 변화를 구분할 수 있는 상향 및 하향 코드값이 생성되어 제안한 방법으로 면분할을 할 수 있었다.

그러므로 횡방향과 종방향으로 적용한 결과의 코드에 의하여 객체의 단위 요소면 별 분할이 수행되었다. 그러나 건물 일부의 외곽선에서는 주위의 식생 등 비건물 데이터에 의해 불규칙한 체인코드가 생성되었다(표 6 참조).



(1) 횡방향 적용 (a) 단순 박공형 건물 (2) 종방향 적용 (c) 계단 박공형 건물



(1) 횡방향 적용



(2) 종방향 적용

(b) 계단 박공형 건물

그림 13. 체인코드에 의한 면분할 결과

표 6. 건물별 패치에 대한 체인코드

건물	횡방향		종방향	
	패치	체인코드	패치	체인코드
(a)	ⓐ	19	ⓒ	7
	ⓑ	10		
(b)	ⓐ	1	ⓓ	25
	ⓑ	1		
	ⓒ	1	ⓔ	16

5. 결론

본 연구에서 model key point와 point cloud 데이터에 3D 체인코드 적용방법을 제시하였다. 실험을 통하여 객체의 형태모사 및 특성분석과 코드로 변환된 객체를 정형화하고 분류하여 객체인식과 모델링에 활용할 수 있음을 보여

주었으며, 결론과 문제점 및 향후 연구는 다음과 같다.

- (1) 3D 체인코드를 이용하여 다양한 형태의 객체를 회전에 불변하게 모사하고 특성을 파악하였으며, 객체 모델링을 위해 코드의 패턴을 분석하여 객체를 정형화하고 분류하여 모델 데이터베이스를 구축할 수 있는 기초자료로 이용할 수 있을 것으로 판단된다.
- (2) 객체의 기울기가 완만하거나 가파르면 기울기의 범위에 의해 결정되는 평면이나 수직면으로 인식되는 문제점이 발생하여 모델링의 결과가 다른 형태로 인식될 수 있다.
- (3) 체인코드로 표현할 수 있는 객체의 방향성은 한계가 있으므로 점밀도 또는 경사의 분포형태에 따라 적층적으로 큐브의 크기를 변화시킬 수 있는 계층적 다중 3D 체인코드에 대한 연구가 필요하다고 사료된다.
- (4) 객체의 형태인식과 모사를 위하여 3D 체인코드 알고리즘을 개선하여 객체 형태를 판별하고 모델링을 위한 정형화와 일반화 연구가 필요하다.

감사의 글

이 논문은 공간정보 전문인력 양성사업의 지원을 받아 수행된 연구임.

참고문헌

김성준, 이임평 (2010), 라이다데이터 분할 알고리즘의 시물레이션 기반 성능평가, 한국지형공간정보학회지, 한국지형공간정보학회, 제 18권 제 2호, pp. 119-129.

박소영, 이동천 (2011), 3D 체인코드에 의한 형태모사와 분석, 한국측량학회 춘계학술발표회 논문집, 한국측량학회, pp. 135-136.

송낙현, 신성용, 조홍범, 조우석 (2007), LiDAR 데이터를 이용한 옥트리 분할 기반의 지붕요소 자동추출, 한국측량학회지, 한국측량학회, 제 25권 제 4호, pp. 327-336.

이선근, 이동천, 염재홍, 임새봄, 김계림 (2007), 수치지도 등고선의 Model Key Point 추출과 Progressive Sampling에 의한 수치지형모델 생성, 한국측량학회지, 한국측량학회, 제 25권 제 6-2호, pp. 645-651.

이진형, 이동천 (2010), 항공영상에 의한 라이다 데이터 분할에 기반한 건물 모델링, 한국측량학회지, 한국측량학회, 제 28권 제 1호, pp. 47-56.

한창호, 오준석, 최병욱 (2010), 3차원 체인코드와 은닉마

- 르코프 모델을 이용한 권투모션 인식, 제어·로봇·시스템학회논문지, 제어·로봇·시스템학회, 제 16권 제 8호, pp. 756-760.
- Blake, A. and Troscianko. T. (1990), AI and the Eye, John Wiley & Sons, New York, NY, p. 290.
- Bribiesca E. (2000), A Chain Code for Representing 3D Curves, *Pattern Recognition*, Vol 33, pp. 755-765.
- Bribiesca E. (2004), 3D-Curve Representation by Means of a Binary Chain Code, *Mathematical and Computer Modelling*, 40, pp. 285-295.
- Haralick, R. and Shapiro, L. (1992), Computer and Robot Vision -Volume 1, Addison-Wesley Pub., Reading, MA., pp. 371-452.
- Lehar, S. (2002) The World in Your Head: A Gestalt View of the Mechanism of Conscious Experience, Taylor & Francis, New York, NY, pp. 52-55.
- Marr, D. (1980) Vision: A Computational Investigation into the Human Representation and Processing of Visual Information, W.H. Freeman and Company, San Francisco, CA, p. 397.
- Schenk, T. (1999) Digital Photogrammetry Vol. 1: Background, Fundamentals, Automatic Orientation Procedures, TerraScience, Laurelville, OH, p. 428.
- Sonka, M., Hlavac, V., and Boyle, R. (1999) Image Processing, Analysis, and Machine Vision, 2nd ed., PWS Publishing, New York, NY, p. 770.
- Tanimoto, S. (1987) The Elements of Artificial Intelligence: An Introduction Using LISP, Computer Science Press, Rockville, MD, p. 527.
- Wulandhari, L. and Haron, H. (2008) The Evolution and Trend of Chain Code Scheme. *International Congress for Global Science and Technology - Graphics, Vision, and Image Processing*, Volume 8, Issue III, pp. 17-23.

(접수일 2011. 08. 09, 심사일 2011. 09. 16, 심사완료일 2011. 09. 26)