

컴퓨터 그래픽스를 이용한 경관시뮬레이션에 있어서 지형상의 구조물 형상 입력과 가시화 방법에 관한 연구

조 동 범

전남대학교 조경학과

A Study on the Data Input and Visualization of Structural Form on Topographic Relief in the Landscape Simulation Technique using CG

Cho, Tong-Buhm

Department of Landscape Architecture, Chonnam National University

ABSTRACT

The purposes of this study were to develop some techniques which can be used in the landscape simulation process using PC based computer graphics. As a result, a couple of utilities were programmed in AutoLISP language. The one(DSLINE.LSP) is to digitize 2-dimensional structure forms in the interactive mode considering error handling, and the other one (IMPOST.LSP) is for superimposing and visualizing the digitized plan data to 3-dimensional solids & surfaces referring to topographic elevations of meshes in digital terrain model. By applying utilities to present site, the followings may be described.

1) The utility DSLINE.LSP for digitizing simplified building structure form were proved to be easy to input data of polygons including orthogonal edges by handling user coordinates system and checking invalid intersection and default closing.

2) IMPOST.LSP utility for superimposing and visualizing tool were proved to be more complicated and speedy in calculating process compared with a existing module of LANDCADD.

3) Utilities developed through this study were proposed to make a practical application of modeling tool before rendering process in landscape simulation of built environment on topographic relief, on specially mesospace level of assessment.

I. 서론

경관시물레이션은 기존 경관현상 및 계획, 설계내용을 전달하는 수단이라는 점에서, 그 기법 적용시 데이터의 적절한 취급에 근거하는 精度 높은 可視化를 전제로 한다. 이러한 점에서 CAD와 Computer Graphics는 資料操作性이라는 면에서 높은 精度 유지가 가능하다는 점(環境科學センタ, 1990) 때문에 그 활용범위가 넓다. 특히 건축이나 순수 도시경관과 같이, 비교적 규칙적 자료를 포함하는 제한된 공간범위에서는 자료조작성을 용이하게 확보할 수 있다. 그러나, 자연경관을 대상으로 하는 경관시물레이션에서는 포함되는 요소(지형기복, 수목 및 다수의 인공구조물 등)가 많고 불규칙적이며, 공간한정성이 적기 때문에 高精度의 자료조작성을 유지하기 어렵게 된다. 오히려 개별적 요소의 정밀도는 낮추더라도 경관요소간의 資料整合성을 유지하는 것이 전체적으로 정도 높은 가시화를 얻는데 유리할 수도 있다(趙, 1994). 특히 지형자료를 포함하는 시물레이션에서는 다른 요소들도 기복있는 표면에 입지시켜야 하기 때문에 정밀도 높은 시물레이션을 위해서는 膨大한 데이터를 작성해야 하고, 그에 따라 처리용량이 증대되어 실용성에까지 영향을 미치게 된다.

지형을 포함하는 축척도면의 건물 형상을 가시화를 위해 평면 데이터를 입력하는 방법으로서 手作業에 의한 것과 도형인식에 의한 것으로 나눌 수 있다. 수작업에 의한 방법은 건물 1棟씩을 頂點座標로써 입력하는 방법이며, 도형인식에 의한 방법은 컴퓨터를 인공지능적으로 사용, 지식베이스의 도형인식기술에 의해 데이터를 자동으로 추출하는 방법(堀, 1987; 鈴木, 1986)이다. 이 경우 자동으로 도형을 추출한다고는 하지만, 래스터화상을 베이스로 윤곽선을 벡터라이징하기 때문에 도형내에서 인식되지 않는 부분이 잔존한다는 문제점을 갖고 있어, 결과를 재확인하는 후처리작업이 요구되는 것이 일반적이다. 결국 수작업에 의한 경우는 入力精度를 유지할 수는 있지만 많은

수의 입력이 필요하다는 점, 자동인식은 사람이 필요없는 완전한 인식은 불가능하다는 점이 각각 단점으로 거론된다. 이러한 단점을 보완하기 위해 도형인식을 이용하되 추출이 제대로 이루어졌는지 사용자가 확인하는 과정을 줄이고 층수데이터를 데이터베이스화하여 많은 수의 건물을 작성하는 방법(間瀬, 1989)이 이용된 바 있지만, 지형데이터를 고려하지 않고 평탄지상의 데이터를 상정하고 있기 때문에 경관시물레이션에서는 부적합하다. 특히 외부로부터의 자료변환과정없이, 지형시물레이션과 구조물 입력, 3차원 자료전환이 동일환경에서 이루어지고 렌더링등 화상작성과정으로 직접 이어질 수 있다는 것은 작업효율상 유리할 뿐만 아니라, 실용성이 관건인 PC base의 시물레이션에서는 중요한 목표가 되기도 한다.

본 연구는 이러한 점에 주목하여, 구조물 형상의 입력방법과 지형모델 표면에 구조물을 위치시키고자 할 때의 지형자료의 처리방법에 대해 다루고자 하였다. 또한 경관시물레이션에서의 폭넓은 활용을 위해, PC base의 작업효율성과 처리고속화에 주안점을 둔 유틸리티 개발을 내용으로 하였다.

II. 연구내용 및 방법

1. 연구의 범위

경관시물레이션에 있어서는 지형기복 뿐만 아니라 수목과 같은 복잡한 요소도 포함되어야 하지만 여기에서는 지형기복만을 기본조건으로 하여 그 표고면에 구조물(건축물을 위주로 하는) 자료가 도입되는 조건에 한정하였다. 건축물이 지형상에 폭넓게 입지하는 상황이라면 건축물과 지형이 상세에 이르기까지 가시화되는 地点的 수준(근경)이 아닌, 地區的 수준(중경 이상)의 조건에서 경관조망이 이루어지는 경우로 보아야 한다. 따라서, 건축물별 화사드 상세 보다는 건축물의 입면규모(층수)와 지붕형태를 주요한 연구범위로 하였다. 본 연구에서는 실용성을 고려하여 PC

base에서의 사용을 전제로 하였고, 컴퓨터 응용 설계에서 범용으로 사용되는 AutoCAD에서의 적용을 위해 LISP language를 유틸리티 개발언어로 하였다.

2. 수치지형자료의 가시화

지형시물레이션에서 보편적인 매쉬형 DTM 이용시 공간적 범위 및 시물레이션 목적에 따라 매쉬 규모가 정해지고 그에 따라 기본자료의 精度가 정해지지만, 가장 중요한 요인으로서는 경관재현 정도의 효율성이 기준이 된다(東海林, 1995). 여기에서처럼 지형상에 다수의 구조물(건물형상)이 포함된 지구적 수준 이상의 경관시물레이션에서는 지형경관 재현성 보다는 건축물의 최소규모가 확인될 수 있는 자료를 지형시물레이션의 기본자료로 설정하는 것이 바람직하다고 할 수 있다. 1/5,000 축척자료를 기본으로 하여도 지구적 수준의 경관재현성면에서는 무리는 없지만 이 경우 건축물 형상을 정확하게 확인할 수 없기 때문에 최소 1/2,500 축척의 항축도(등고선간격 1미터)를 기본자료로 하는 것이 바람직하다. 본 연구의 프로그램 시행을 위해 적용한 사례 대상지는 항축에 의한 GIS 자료가 기구축된 지역으로서, 등고선 및 TIN 데이터는 3차원 자료로 기작성되어 있었기 때문에 입수된 전산자료로부터 3차원 자료만을 추출가공하여 DTM을 작성하였다.

3. 구조물 평면 형상의 입력방법

가. 구조물 자료의 취급범위

경관자료의 데이터베이스화라는 면에서 의미를 부여할 수 있고, 비교적 규칙성 있는 구조물에 한정하기 위해 건물자료만을 대상으로 하였고 기타 요소(도로등의 토목구조물)는 제외하였다. 기구축된 GIS 자료를 그대로 이용하는 경우, 건축물 형상은 일반적으로 2차원

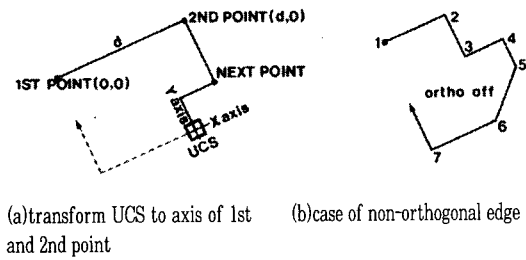
자료로 작성되어 있으며 입력상 일관성이 없는 경우(건축물의 평면형상이 폐합되어 있지 않은 경우라든지, line entity와 pline을 혼용하는 경우 등)도 발견할 수 있기 때문에, 자료변환만으로 건축물 형상을 가시화하기는 곤란하다. 도면자료를 스캐닝한 래스터 화상자료로부터 건축물 외곽선만을 벡터라이징하는 방법도 별도의 하드웨어와 소프트웨어를 필요로 하며, 精度 높고 복잡한 자료를 스스로 할 경우 PC base의 작업에서는 비실용적이라 할 수 있다. 또한 원자료가 완전하지 않은 경우(폐합되지 못한 상태나 모서리 부분의 형태가 일관성 없는) 후처리를 요하기 때문에 번거롭다는 점도 생각할 수 있다. 따라서 여기에서는 건축물의 평면 형상을 충분히 파악할 수 있는 자료를 스스로 이를 도면으로 출력한 후 디지털라이징하는 방법을 선택하였다. 이 경우 추가입력이 가능하다는 점을 장점을 들 수 있는데, 건물이 밀집된 도시지역의 경우 데이터가 빈번히 경신되어 간다는 점 때문에 축도된 기본자료에 데이터를 추가한다는 것은 매우 중요한 문제라 할 수 있다.

나. 건축물 평면형상의 특징과 입력방법

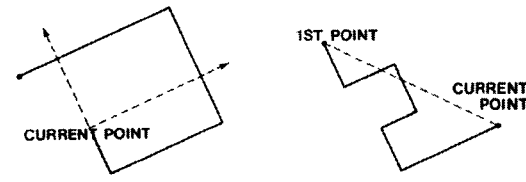
건축물의 평면형상은 직각을 이루는 모서리로 이루어지는 경우가 많고, 직각이 아닌 경우라 하더라도 직각형상의 일부분에 포함되는 경우가 일반적이므로, 결국 직각모서리들로 이루어진 폐합다각형의 효율적인 입력방법이 요구된다고 할 수 있다. 축척도면상에서 건물 외곽선을 기존의 일반적 방법으로 디지털라이징하는 경우, 작은 도형을 대상으로 한 수작업이기 때문에 직각을 유지해야할 모서리가 뒤틀리게 된다는 상황을 피할 수 없게 된다. 여기에서는 최초로 시작하는 점을 사용자 좌표시스템의 원점으로 하고 그 점과 두번째 디지털라이징된 점을 연결하는 방향을 사용자 좌표시스템의 축에 일치시킨 후, 직각모드를 유지¹⁾하면서 이후의

1) AutoLISP상에서는 orthomode = 1

정점들을 입력하도록 하여 이와 같은 문제점을 해결하였다(그림 1의 (a) 및 프로그램 1에서 (1)로 표시된 부분이하 16라인의 주요 내용).



(a)transform UCS to axis of 1st and 2nd point (b)case of non-orthogonal edge



(c) checking invalid intersections and default closings

Figure 1. Digitizing mode setting and error checking concepts in PROGRAM 1.

사용자는 각기 다른 방향의 건물에 대해 사용자 좌표계 변환없이 디지털링만을 수행하고 프로그램상에서 이를 자동인식 처리하게 되므로 효율적인 입력이 가능하였다. 한 건물형상내에 부분적으로 직각이 아닌 모서리를 포함하는 경우에는(그림 1의 b) 직각모드를 해제한 후 진행하고 다시 설정하게 된다. 건물형태에서는 외곽선간 교차하는 경우가 없으므로 이러한 입력 오류가 발생한다든가, 현재까지 입력된 점에서 폐합하고자 할 때 최초 시작점을 연결하는 선이 다른 선들과 교차될 상황의 경우에는 이를 자동으로 인식하여(그림 1의 c 및 프로그램 1의 (2)로 표시된 부분 이하 39라인의 주요내용) 입력 오류가 발생하지 않도록 하였다.

다. 건축물 입면 형상의 단순 모델화

경관평가를 위한 예측화상을 작성하는 경우, 경관평가의 목적이나 評價視點에 따라서는 입력

수고나 기억용량등의 문제도 있기 때문에 실제의 모든 정보를 입력하는 것은 거의 불가능하며 어느 정도 생략될 필요가 있다. 이를 위해 입력 대상이 된 건축물을 대상으로 하여 간략표현할 수 있는 모델화를 전제로 하였다. 여기에서 문제가 되는 것은 건축물의 입면화사드에 부가된 상세 형태와 한 건물내에서 부분적으로 입면규모가 다른 경우 및 지붕부분의 형태이다. 본 연구에서는 앞서 전제로 한 바와 같이 근경에서의 상세한 화상을 목적으로 한 것이기 아니므로, 1동의 건물을 표현하기 위한 기본적인 요소로서 「외벽」과 「지붕」만을 대상으로 하고 (그림 2)와 같은 모델을 설정하였다.

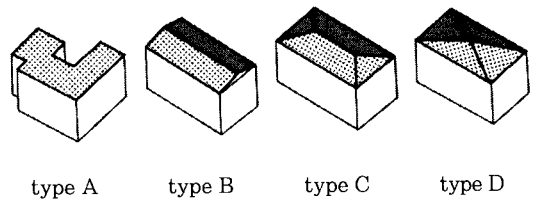


Figure 2. Simplified types in modeling of building.

- ①어떠한 건물이든 외벽과 지붕, 두가지 요소로 구성되며, 처마의 돌출은 없는 것으로 가정하였다.
- ②외벽과 지붕에는 창과 출입문, 기타 재료는 부가되지 않는 데미표면으로 하였다.
- ③어떠한 건물이든 층간의 형태차이는 없고 1층(최초 디지털링된 형태)과 같은 형태를 가지며 층간의 층수 차이는 없는 것으로 가정하였다(즉, 1층부와 2층부가 한 건물로 그려진 경우에는 2층으로 보았다).
- ④지붕의 종류로는 A, B, C, D의 4가지 유형을 설정하였으며, 평면 형태가 4각이 아닌 경우에는 평면형 지붕(A형)만으로 가정하였다.
- ⑤지붕의 최고지점과 외벽부의 높이차는 일정한 것(2.0미터)으로 보았으며, B형과 C형은 건물의 장변방향과 지붕선이 일치하는 것으로 가정하였다.

건물 평면형상의 디지털링 프로그램에서 정점 연결이 완료되면 층수와 지붕유형을 순차적

으로 입력요구하여 데이터화 한다.

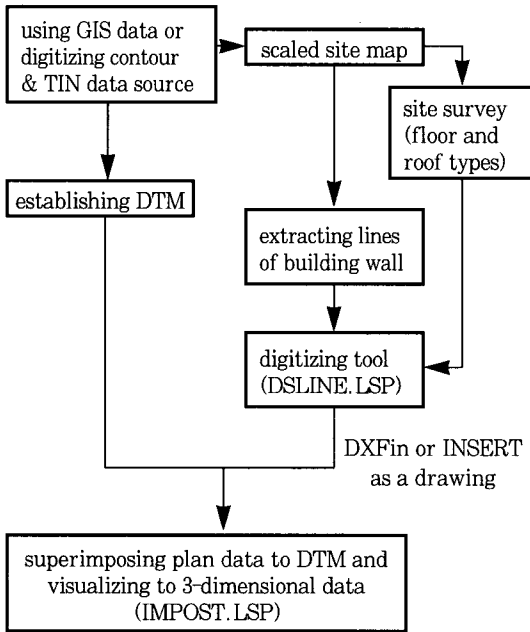


Figure 3. Utility application process.

향측도상에 건물의 층수와 구조가 기입되어 있는 수도 있지만 향측도의 작성 목적에 따라서는 누락되어 있는 수가 많기 때문에 별도의 현지조사가 필요하다. 본 연구에서 프로그램의 試行을 위해 입수한 자료상에는 층수가 기재되어 있지 않아 층수 및 지붕유형은 현지조사에 의해 이루어졌다. 이상의 구조물 평면형상의 입력과정 및 이후 과정으로의 적용은 <그림 3>과 같다.

4. 지형과 건물자료 정합 및 가시화

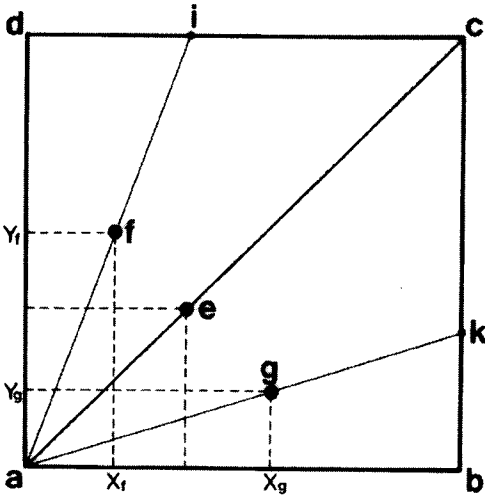
가. 수치지형자료에의 건물자료 도입방법

프로그램 1에서 디지털이징된 건물자료는 DXF 화일을 경유하거나 도면삽입에 의해 수치지형자료와 하나의 도면환경으로 일치시키게 된

다. 프로그램 실행의 편의상 건물자료는 하나의 레이어를 차지하고 있으며 DTM의 매쉬자료도 별도의 독립된 레이어에 보관되도록 하였다. 건물자료는 2차원의 평면자료로서, 1동별로 가장 마지막에 데이터베이스에 추가된 층수와 지붕형태 데이터가 텍스트 데이터로 들어오게 되고 순차적으로 각변의 정점자료가 line entity로서 나열되는 구조를 갖고 있으므로 선택집합 데이터의 순차적 구조를 이용하여 건물별 자료 구분이 가능하도록 하였다. 건물을 지형상에 입지시키기 위해서는 지형을 건물형태에 맞추어 평탄화하는 방법을 먼저 생각할 수 있지만, 여기에서와 같이 지형자료가 매쉬형태의 사각형구조일 경우 건물주변의 부분에 이르기까지 지형이 변경되는 셈이어서 인접한 다른 건물의 표고결정에 영향을 주게 되며, 또한 DTM 작성의 소스자료가 건물을 포함한 지형이기 때문에 지형의 변경없이 그대로 건물의 표고가 결정되는 것이 바람직하다. 이를 위해서는 지형기복과 건물바닥면 사이에 가능한 이격이 없는 조건을 상정하여야 한다. 건물의 어느 정점에 맞추어 지형표고만을 계산하면 기복있는 지형상에서는 건물의 일부분이 지표면과 분리되는 경우가 경우가 발생하기 때문이다.

본 연구에서는 DTM 수치지형자료의 계산을 위해 LANDCADD의 Quadrangle 모듈을 활용하였는데 여기에도 지형상으로 3차원의 요소를 입지시키는 기능²⁾이 존재한다. 그러나 이 기능은 수목등의 3차원 도형블럭을 DTM내 평면위치에 삽입한 후 지형상의 위치로 표고조정하기 위한 것으로서, 수목은 근원부(도형삽입점의 점 좌표)로 지형상의 위치를 결정하기 때문에 기준점이 단일인 경우에는 유효하지만, 건물과 같이 기준점이 없이 면으로 볼 수 있는 경우에는 적용이 곤란하다. 또한 line entity의 경우 두개의 정점이 매쉬의 3DFACE와 교차할 때만 유효하고 한 매쉬내에 두개의 정점이 존재하여 매쉬와 교차점이 생기지 않는 경우 벡터에러가 유발된다는 결점

2) LANDCADD menu 상으로는 "Adjust Elevation" 및 "Superimpose"에 해당된다.



$$Z_e = Z_a + D_{ae} / (\sqrt{2} MS \times (Z_c - Z_a)) \text{ ----- (1)}$$

$$Z_f = Z_a + D_{af} / D_{af} \times (Z_i - Z_b) \text{ ----- (2)}$$

$$Z_g = Z_a + D_{ag} / D_{ak} \times (Z_k - Z_a) \text{ ----- (3)}$$

$$Z_i = Z_d + D_{di} / MS \times (Z_c - Z_d)$$

$$D_{di} = (X_f + X_a) \times (Y_f - Y_a) / MS$$

where,

a~d : 4 vertex of mesh as a 3DFACE entity

MS : unit size of mesh

Z_e, Z_f, Z_g : elevations of point a~d

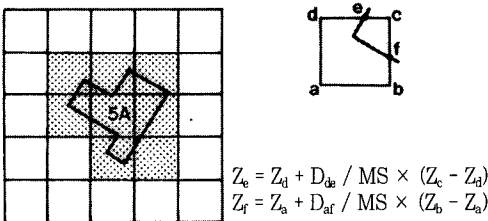
D : distance between two points in XY plane

Figure 4. Decision methods of elevations(z values) in vertex of building polygons.

을 갖고 있다. 여기에서는 가능한한 지형집합부의 표고계산 精度를 높이기 위해 건물의 각 정점 및 매쉬 격자선과 교차되는 점들의 표고를 계산하고 그 중 최소치를 건물 입지표고로 하였다. 각 정점위치가 포함될 매쉬표고의 규정(그림 4)은 각 정점좌표(2차원 데이터)로부터 매쉬 첫 정점³⁾의 2차원 데이터를 찾고, 이를 이용 매쉬자료의 선택집합 데이터베이스에서 filtering하므로서 해당 매쉬의 4개 정점

좌표를 읽어내는 방식을 취하였다(프로그램 2에서 (1)로 표시된 부분 이하 20 라인). 건물의 각 정점 위치는 매쉬의 제 1정점과 제 3정점을 잇는 대각선상(e, 그림 4 (1)), 대각선 위쪽(f, 그림 4 (2)), 아래쪽(g, 그림 4 (3)) 등 그 위치에 따라 계산 방식이 정해지는데, 이는 최종적으로 매쉬면의 렌더링 방식이 이 대각선에 의해 분할된 3각면을 하나의 평면으로 정하여 이루어 진다는 점에 기초하였다.

매쉬와 건물의곽선의 교점표고는, 건물이 포함되는 매쉬 구간을 x축, y축별로 나누어 평면상 교점을 추출하고, 이를 포함하는 해당매쉬를 데이터베이스에서 filtering하여 3DFACE 두 정점간의 표고차로서 계산하였다(그림 5).



where,

a~d : 4 vertex of mesh as a 3DFACE entity

$Z_a \sim Z_d$: elevations of point a~d

MS : unit size of mesh

D : distance between two points in XY plane

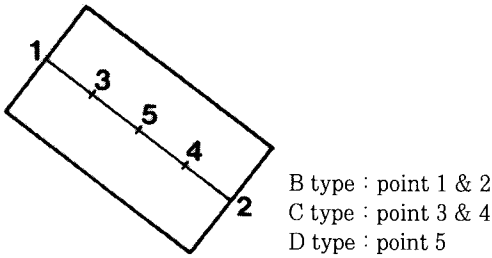
Figure 5. Decision method of intersecting points by meshes and building line in XY plane.

나. 건물입면 및 지붕형태의 3차원 자료화

각 건물의 입지위치(표고)가 결정되면 0 표고에서 각 표고로 데이터베이스를 경신한 후 이미 변수로 저장된 층수데이터로 두께(1층의 높이는 3.0미터로 상정)를 주어 외벽면을 완성하였다. 지붕형은 입력자료에 따라 4가지 유형으로 모델

3) 3DFACE의 제 1 정점(左下 정점) : 엔티티 데이터베이스상에서는 association 10과 관련된 리스트.

링하였는데, 평면형을 제외한 B, C, D형은 건물 평면형태가 4각형인 경우에만 한정하였다. 각 유형에 있어 지붕정점의 결정방법은 <그림 6>에 나타낸 바와 같으며, 지붕의 높이는 일정한 것으로(2.0미터) 상정하였다.



where, point 1 is midpoints of point a & b, point 5 is midpoint of point 1 & 2, and so on.

Figure 6. Dicisions of highest points in each roof types of B, C, and D.

지붕의 각면은 3DFACE 엔티티로서 건물의 방향에 따라 자동 작성되어 데이터베이스에 추가되도록 하였는데, 평면형은 비교적 단순형태일 때 3DFACE로도 가능하지만 복잡한 다각형인 경우에는 면분할이 복잡해져 solmesh를 적용하여 surface modeling되도록 하였다. 외벽 및 지붕표면을 표면모델링으로 처리한 것은 렌더링시 색채나 재질감을 부여하기 위한 전처리 측면도 있다. 이를 위해 건물의 층수별, 지붕유형별 레이어로 각 엔티티 데이터베이스가 경신되도록 하였다. 각 지붕유형에 따른 데이터베이스에의 엔티티 생성은 프로그램 2에서 (2)로 표시된 부분 이하의 내용이다.

Ⅲ. 결과 및 고찰

1. 건물자료 입력 유틸리티의 개발

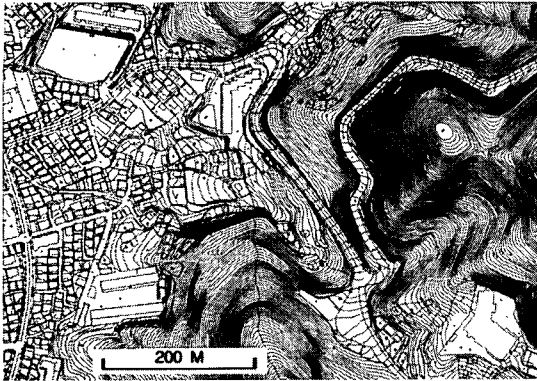
축척도면상 건물의 자료화를 위해 본 연구에서 취한 방법은 기본적으로는 수작업에 의한 입력에 해당되지만, 가능한 건물형상의 일반적

인 특징을 이용하여 건물 하나하나를 모델링하고 위치, 회전변환하여야 하는 기존 작업의 번거로움을 해결하였다는 점 및 범용의 CAD, CG 유틸리티와 연계될 수 있도록 하므로써 실용성을 확보한 점에서 특징을 찾을 수 있다. 지리정보시스템에서 등고선등의 입력시 빈번하게 사용되는 화상처리방법(스캐닝-벡터라이징)도 유용성을 가지지만, 층수와 지붕형의 자료가 최초 원자료로부터 주어져야 하고 이에 따른 처리작업이 추가적으로 요구된다는 점 때문에 기존 유틸리티 그대로의 활용은 제한된다. 또한 원자료의 스캐닝 입력 해상도에 따라 소규모의 건물은 처리상 제한된다는 점도 무시할 수 없다고 보여진다.

본 연구에서는 입력 건물의 처리정도는 디지털타이징할 도면의 축척에 의해 결정되는 셈인데, 건축물의 평면형을 확인하기 위해서는 최소한 축척도 기반의 1/2,500 축척 정도가 필요할 것으로 판단되었다. 소규모 구조물로만 이루어진 경우 소축척으로 출력하여 입력할 수도 있지만, 이 경우에는 입력기기의 제한을 받기 때문에 도면분할이나 대형 디지털타이저를 이용하여야 하므로 하드웨어 구성의 제한이 따른다. 일반적인 소형 디지털타이저(12"×12"나 12"×18"형)를 이용하는 경우, 도면 분할없이 입력할 수 있는 범위는 1/2,500 축척에서 약 750x750m~1000m에 불과하지만 지구적 수준의 경관예측은 가능하다고 볼 수 있다. 프로그램의 試行을 위해 실제 대상지(그림 7의 (a), 700×500m 규모)내의 건물형상만을 추출, 디지털타이징한 결과가 <그림 7>의 (b)이다.

2. 지형자료와의 정합 및 3차원 자료화

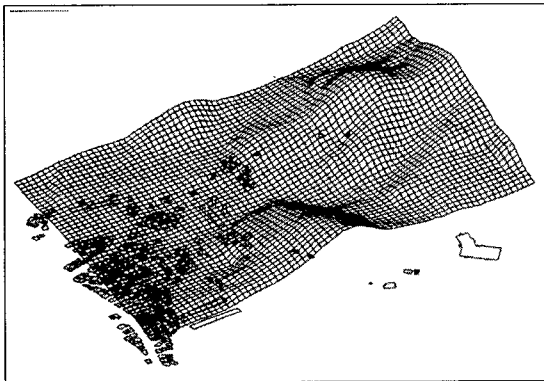
수치지형을 나타내는 매쉬표면에 구조물의 평면형태를 입지시키는 경우, 와이어프레임만의 시뮬레이션에 한정된다면 건물이 입지하는 매쉬들의 평균 표고를 건물의 지반고로 보아도 큰 문제점은 드러나지 않게 된다. 그러나 와이어프레임 검토에 그치지 않고 표면렌더링에 의해 색채나 재질감까지 부여되어야 할 상황이라



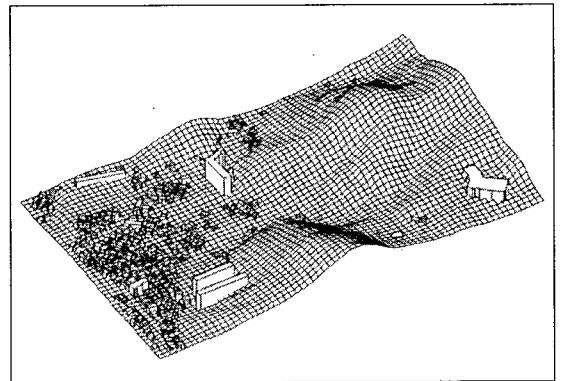
(a) Source map of existing site



(b) digitized plan by DSLINE.LSP



(c) before applying IMPOST.LSP



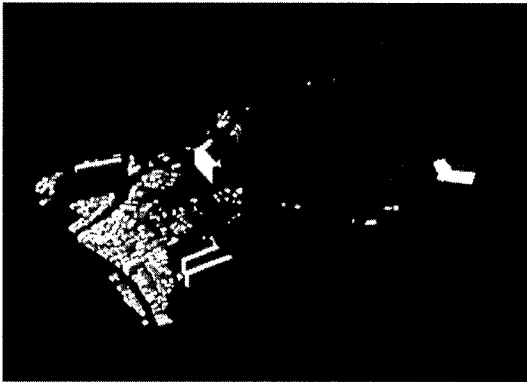
(d) after applying IMPOST.LSP

Figure 7. Utilites application case to a existing site.

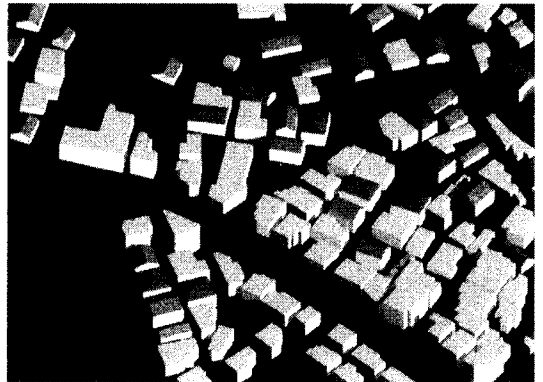
면 매쉬크기에 의해 결정되는 기복의 미세함과 건물 底面의 정합은 최종적인 시뮬레이션 화상의 정확성에도 영향을 주게 되므로 자료처리상 중요한 관건이 되는 셈이다. 건물은 비교적 평면상에 입지되는 것이 일반적이며, 구릉지에서는 성, 절토가 이루어져 복잡한 인접지형의 변경이 이루어지기 때문에 건물이 입지한 후 매쉬지형을 변경하여 삼각폴리곤으로 분할하는 방법(紫崎, 1991)도 바람직하다. 그러나 많은 수의 건물이 근접하여 배치되는 경우에는 폴리곤의 수가 많아지고, 건물규정 데이터의 추가 이상으로 처리용량이 증가된다. 따라서 건물 주변의 미세한 기복이나 건물지하부를 고려하지 않고 입면을 위주로 하는 시뮬레이션이라면 지형을 그대로 존속시킨채 건물데이터를 추가하

는 것이 바람직하다고 할 수 있다.

본 연구에서와 같이 지형과 건물을 정합시키는 방식일 경우 지형과 건물이 이격되지 않는다는 장점이 있는 반면 건물을 포함하는 매쉬가 기복이 있을 때 건물지면 일부분은 지표면 내에 포함된다는 단점을 갖는 것으로 나타났다. 그러나 건물이 이미 입지되어 있는 지형자료로부터 작성된 매쉬라면 매쉬단위 규모가 건물규모에 비해 상대적으로 월등히 크지 않는 한 그 오차는 크지 않은 것으로 볼 수 있었다. 또한 건물지반고를 일정한 표고로 구하고 건물과 교차하는 매쉬단위를 건물지반고에 맞출 수도 있지만 이 경우 매쉬의 4정점이 서로 연결되어 있기 때문에 주변의 다른 매쉬표고까지 변화된다는 점에서 곤란할 것으로 생각되었다.



(a) rendered perspective image



(b) detailed view of building's wall and shadow on landform relief

Figure 8. Final rendered images of visualized existing landscape.

사례로 제시된 시뮬레이션 결과(그림 8)는 지구적 규모의 경관을 대상으로 특정 시점을 설정하지 않고 대상지 전체 건물을 같은 精度로 하였기 때문에 근경과 원경의 건물의 상세차이가 없었다. 사전에 예측시점이 선정되어 있다면, 근경 일부는 벽면상세를 별도로 모델링하고 렌더링시 시거리에 근거하여 DTM상에 위계적 매핑(Kaneda, 1989)을 적용하므로써 원근경의 상세차이에 의한 투시적 효과를 높일 수 있으리라 보여진다. 또한 지붕의 형태도 단순화한 면이 있지만 추가적인 유형 설정의 여지가 있으며, 자료처리후에도 작업량이 많지 않은 경우에는 일부 형태 附加(예를 들면 고층 아파트의 엘리베이터층 등)를 독립적으로 적용할 수 있으리라 보여진다.

직접적인 비교는 어렵지만, 외부적인 데이터 처리과정(매쉬 및 건물 line 엔티티를 DXF 화일로 출력, 자료화한 후 자료들간 참조계산 결과를 다시 DXF화일 경신에 반영하는 방법)의 번거로움에 비하면 처리속도도 상대적으로 빠른 것으로 볼 수 있었다. <그림 7>의 (c)와 (d)는 지형과 건물이 정합되기 전후 상황을 나타낸 것이다. 사례 DTM의 매쉬규모는 10미터 자료였지만 유틸리티에서는 매쉬규모를 DTM 자료로부터 인식하도록 하였기 때문에 처리상 제한은 없다. <그림 8>은 건물과 지형과의 정합성 효과를 알아보기 위해 지형상으로

이행된 결과를 표면렌더링에 연결시킨 畫像이다. 건물의 음영부분에 주목하면 건물과 지형간에는 이격이 발생하지 않고 있음을 확인할 수 있었다.

3. 연구결과의 활용성과 제한성

1) 도시경관평가에의 활용성 : 도시경관의 경우에는 규모와 형태를 달리하는 다수의 건축물이 존재하면서도 평가목적에 따라 제한된 시점으로부터의 장면경관만 다루기 어려우며, 보다 공간적 범위가 넓은, 場으로서의 경관이나 도로 등으로부터 지각되는 연속적 경관이 평가대상이 될 필요성이 있다. 이를 위해서는 넓은 범위의 경관요소를 자료화하고 가시화하여야 하기 때문에 자료의 膨大함과 함께 자료조작의 충실성을 기하기 어렵다. 본 연구에서의 방법은 데이터 처리의 고속화라는 측면에서 이러한 점을 충족시킬 수 있을 것으로 판단되지만 세부형상까지는 고려하지 않았기 때문에 도시스카이라인의 검토나 가로경관의 볼륨감 검토 측면에서는 충분한 활용성을 가질 것으로 보인다.

2) 자연경관평가에의 활용성 : 자연경관의 경우에는 지형이나 기타 복잡한 형상의 경관요소가 다수 포함되어 있기 때문에 이들 자료의 중요성을 강조한다면 구조물등의 부가적인 자

료를 도입하는 경우 원자료의 변경없이 임의적으로, 즉 서로 독립적이면서도 정합성있는 자료구성이 가능하여야 한다. 구릉지등의 자연지형상에 구조물등이 다수 존재하는 상황이 그 예가 되며, 녹지경관에 대한 건축물의 경관적 영향과 관련하여 이루어지는 예측이나 평가(광주광역시, 1996)시 활용될 수 있는 기법중 하나로서 활용성을 가진다.

3) 경관가시화 기법으로서의 제한성 : 본 연구의 자료 입력부분에서는 수작업에 관한 효율성 제고만을 다루었기 때문에 작업자에 따라 나타날 수 있는 입력오차등은 해결할 수 없었으며, 이를 위해서는 화상처리에 의한 방법과의 병용이 요구된다. 또한 지붕형태를 몇가지 단순유형으로 제한하므로써 실제 형태를 충분히 반영하는데는 제한이 된다는 점과, 지형상으로의 이행을 위한 표고계산에 있어 매쉬수가 많을수록 매쉬 선택집합의 데이터베이스 참조 및 검색시간이 소요된다는 점 및 복잡한 다각형의 평면지붕면을 생성하기 위해 solmesh 기능을 이용하므로써 처리시간이 지연된다는 점 등이 개선사항으로 남는다.

IV. 결론

지형자료의 처리는 경관시뮬레이션 및 조경계획 설계에 있어 활용범위가 넓다. 그러나 지형모델링이나 해석만으로는 현황, 예측경관의 가시화에 부족하며 지표상의 건축물, 구조물, 수목 등의 자료를 효율적으로 처리할 수 있을 때 그 활용성이 증대될 수 있다. 본 연구에서는 지형상의 건축물 간략입력과 자료처리가 기존 유틸리티에 불비되어 있다는 점에 주목, 효율적인 구조물 입력방법을 제시하고 이에 의한 구축자료를 매쉬지형면

상에 정합시킬 수 있는 일련의 자료변환 알고리즘을 개발하고자 하였다. 이를 지구적 규모의 경관시뮬레이션에 시행해보므로써 유틸리티로서의 유용성을 확인하고자 하였는데, 높은 精度의 CG 기법으로서의 제한이 있었지만 범용의 유틸리티인 AutoCAD와 Landcadd의 환경내에서 보완사용할 수 있다는 점에서 유용성을 찾을 수 있었으며, PC base에서 용이하게 사용할 수 있다는 점에서 실용적일 것으로 판단되었다.

인용 문헌

1. 광주광역시(1996), 광주시 도시경관보전.관리방안에 관한 연구, 광주광역시 : 76-108.
2. 趙東範(1994), 컴퓨터그래픽스 응용에 의한 경관정보의 가시화 연구, 대한국도.도시계획학회지, 29(4) : 311-321.
3. 間瀬實朗, 瀨口哲夫(1989), 都市景觀シミュレーションのための建物形状データ入力支援システムの研究, 日本都市計劃學會學術研究論文集, 24 : 421-426.
4. 堀, 古賀, 恒川(1987), 總合化地理情報システム 3. 圖形認識能力をもった圖形エディタ, 情報處理學會第34回全國大會1E-3, 日本情報處理學會 : 1791.
5. 東海林克彦, 齊藤 馨(1995), メッシュ標高モデルを用いた可視, 不可視領域シミュレーションの再現性に關する調査, 日本造園學會誌ランドスケープ研究, 58(5) : 189-192.
6. 鈴木, 星野, 山田(1986), 地圖認識入力システムMARIS-自動認識處理-, 情報處理學會第33回全國大會4NE-4, 日本情報處理學會 : 1485.
7. 社團法人 環境情報科學センタ編 (1990), 自然環境アセスメント指針, 朝倉書店 : 84-87.
8. 紫崎亮介(1991), 都市地域計劃における地理情報システムの利用に關するワークショップ, 日本測量協會 : 111-123.
9. Kaneda, K. et al.(1989), Three Dimensional Terrain Modeling and Display for Environmental Assessment, ACM SIGGRAPH Computer Graphics, 23(3) : 207-214.
10. Landcadd Reference Manual II(1990), Quadrangle, Landcadd International, Franktown, USA. : 3-1 - 3-34.

APPENDIX

PROGRAM 1 : DSLINE.LSP

```

(princ "loading DSLINE...")(terpri)
;; utility for Digitizing Structure-LINE segments
;; 1996. 7.
;;
;;-----Transform radian to degree -----
(defun rtd (a)
  (/ (* a 180.0) pi)
)
;;----- Main Program -----
(defun C:DSLIN (/ s ort cmd bli fault f rf pnt1 pnt2 pnt3 stpt ntxtpt vpnt d
  e cnt i ndex i tmp pa pb tx ty mnx mxx mny myx edg_ang bd_unit seg )
  ;;
  ;;----- Error Handling Function -----
  (defun clerr(s)
    (if (/= s "Function cancelled")
      (princ (strcat "\nError: "s)))
      (setq er (sslength bd_unit) eri (- er 1))
      (repeat er
        (entdel (ssname bd_unit eri))
        (setq eri (- eri 1))
      );:repeat
      (setvar "cmdecho" 0)
      (setvar "blipmode" 0)
      (setvar "orthomode" 0)
      (command "UCS" "w" )
      (command "LAYER" "s" "0" "")
      (command "ELEV" "0" "0")
      (setq "error" olderr clerr nil)
      (setq bd_unit nil dir nil)
      (princ)
    );:clerr
  ;;
  (setq olderr "error" "error" clerr)
  (setq ort (getvar "orthomode"))
  (setq cmd (getvar "cmdecho"))
  (setq bli (getvar "blipmode"))
  (graphscr)
  (setvar "cmdecho" 0)
  (setvar "blipmode" 0)
  (setvar "orthomode" 0)
  (command "UCS" "w" )
  (command "LAYER" "m" "DIGITLINE_BLDG" "")
  (command "ELEV" "0" "0")
  (setq edg_ang nil dir nil)
  (initget 1)
  (setq pnt1 (getpoint "\nStart point : "))(terpri) ----- (1)
  (command "UCS" "0" pnt1 )
  (setq mnx 0 mny 0 mxx 0)
  (setq pnt1 (list 0.0 0.0))
  (initget 1)
  (setq pnt2 (getpoint pnt1 "\nTo 2nd point : "))(terpri)
  (setq r (rtd (angle pnt1 pnt2)))
  (command "UCS" "Z" r)
  (setq d (distance pnt1 pnt2))
  (setq pnt2 (list d 0.0))
  (setq mxx d fault 0)
  (command "LINE" pnt1 pnt2 "")
  (setq bd_unit (ssadd))
  (setq seg (entlast))
  (ssadd seg bd_unit)
  (setvar "orthomode" 1)
  (while fault
    (initget 1)
    (setq pnt3 (getpoint pnt2 "\nTo 3rd point : "))(terpri)
    (setq e (atof (angtos (angle pnt2 pnt3))))
    (if (or (= e 0) (= e 180)) (setq fault 1) (setq fault 0))
    (if (= fault 1) (princ "\nInvalid digitizing -- Try again!"))
    (if (= fault 0) (setq fault nil))
  );:while fault
  (if (> mny (cadr pnt3)) (setq mny (cadr pnt3)) (setq mny mny))
  (if (< mxy (cadr pnt3)) (setq mxy (cadr pnt3)) (setq mxy mxy))
  (command "LINE" pnt2 pnt3 "")
  (setq seg (entlast))
  (ssadd seg bd_unit)
  (if (or (= e 90) (= e 270))
    (setq edg_ang "RECT") (setq edg_ang "ANGL"))
  (setq stpt pnt3 vpnt pnt2 ntxtpt pnt3)
  (setq cnt 2 fault 0)
  (while ntxtpt ----- (2)
    (setq index (- (sslength bd_unit) 1) fault 0 chk 0)
    (while fault
      (setq i 1 chk 0)
      (repeat (- index i)
        (setq tmp (entget (ssname bd_unit i)))
        (setq pa (list (car (cdr (assoc 10 tmp))) (cadr (cdr (assoc 10 tmp)))))
        (setq pb (list (car (cdr (assoc 11 tmp))) (cadr (cdr (assoc 11 tmp)))))
        (setq pa (trans pa 0 1) pb (trans pb 0 1))
        (if (and (= chk 0) (= (inters pa pb pnt1 stpt) nil))
          (setq chk 0) (setq chk 1))
          (setq i (+ i 1))
        );:repeat
        (if (or (= chk 1) (and (/= dir nil) (or (= e 90) (= e 270))))
          (progn
            (initget 1)
            (setq ntxtpt (getpoint stpt "\nTo next point : "))
            (setq ntxtpt (getpoint stpt "\nTo next point/Close : "))(terpri)
            (if (/= ntxtpt nil) (setq dir (atof (angtos (angle stpt ntxtpt)))))
            (setq i 0 fault 0)
            (if (/= ntxtpt nil)
              (progn
                (repeat index
                  (setq tmp (entget (ssname bd_unit i)))
                  (setq pa (list (car (cdr (assoc 10 tmp)))
                    (cadr (cdr (assoc 10 tmp)))))
                  (cadr (cdr (assoc 10 tmp))))
                  (setq pb (list (car (cdr (assoc 11 tmp)))
                    (cadr (cdr (assoc 11 tmp)))))
                  (cadr (cdr (assoc 11 tmp))))
                  (setq pa (trans pa 0 1) pb (trans pb 0 1))
                  (if (and (= fault 0) (= (inters pa pb stpt ntxtpt) nil))
                    (setq fault 0) (setq fault 1))
                    (setq i (+ i 1))
                  );:repeat
                  (if (= fault 0)
                    (if (or (= (- e dir) 0) (= (abs (- e dir)) 180))
                      (setq fault 1) (setq fault 0))
                      (if (= fault 1)
                        (princ "\nInvalid digitizing -- Try again!") (setq fault nil))
                      );:if
                    (if (/= ntxtpt nil) (setq fault nil))
                  );:while fault
                (if (/= ntxtpt nil)
                  (progn
                    (setq vpnt stpt endpt ntxtpt)
                    (setq e (atof (angtos (angle stpt endpt))))
                    (command "LINE" stpt endpt "")
                    (setq seg (entlast))
                    (ssadd seg bd_unit)
                    (if (= edg_ang "RECT")
                      (if (and (/= e 0) (/= e 90) (/= e 180) (/= e 270))
                        (setq edg_ang "ANGL") (setq edg_ang "RECT"))
                      );:if
                    );:if
                    (if (and (= ntxtpt nil) (= cnt 2))
                      (progn
                        (setq vpnt pnt3)
                        (command "LINE" stpt pnt1 "")
                        (setq seg (entlast))
                        (ssadd seg bd_unit)
                      );:if
                    (if (and (= mnx 0) (= ntxtpt nil)) (setq stpt (list (car pnt1)(cadr vpnt))))
                    (if (> mnx (car stpt)) (setq mnx (car stpt)) (setq mnx mnx))
                    (if (< mnx (car stpt)) (setq mnx (car stpt)) (setq mnx mnx))
                    (if (> mny (cadr vpnt)) (setq mny (cadr vpnt)) (setq mny mny))
                    (if (< mny (cadr vpnt)) (setq mny (cadr vpnt)) (setq mny mny))
                    (setq stpt endpt)
                    (setq cnt (+ cnt 1))
                  );:while ntxtpt
                (if (> cnt 3)
                  (progn
                    (entdel (entlast))
                    (ssdel seg bd_unit)
                  );:if
                (if (and (> cnt 3) (= (cadr vpnt) (cadr stpt)))
                  (progn
                    (command "LINE" vpnt (list (car pnt1) (cadr vpnt)) "")
                    (setq seg (entlast))
                    (ssadd seg bd_unit)
                    (command "LINE" "@@" pnt1 "")
                    (setq seg (entlast))
                    (ssadd seg bd_unit)
                  );:if
                (if (and (> cnt 3) (/= (cadr vpnt) (cadr stpt)))

```

```

(progn
(command "LINE" vpnt (list (car pnt1) (cadr stpt) ""))
(setq seg (entlast))
(ssadd seg bd_unit)
(command "LINE" "@" pnt1 "")
(setq seg (entlast))
(ssadd seg bd_unit)
));:if
(if (< mx (car pnt1)) (setq mx (car pnt1)) (setq mx mx))
(if (> my (cadr vpnt)) (setq my (cadr vpnt)) (setq my my))
(if (< my (cadr vpnt)) (setq my (cadr vpnt)) (setq my my))
(setq lx (/ (+ mx mx) 2))
(setq ly (/ (+ my my) 2))
(setq fault 0)
(setq f (getint "No. of Building Floor/ <1> : "))(terpri)
(if (= f nil) (setq f 1))
(if (and (= cnt 4) (= edg_ang "RECT"))
(while fault
(setq rf (getstring "Roof Type (A/B/C/D) / <A> : ")
(if (or (= (strcase rf) "A") (= (strcase rf) "B")
(= (strcase rf) "C") (= (strcase rf) "D") (= rf ""))
(setq fault nil) (setq fault 0))
);:while
(while fault
(setq rf (getstring "Roof Type(A type only on this case) / <A> : "))(terpri)
(if (or (= (strcase rf) "A") (= rf ""))
(setq fault nil) (setq fault 0))
);:while
);:if
(if (= rf "") (setq rf "A"))
(setq rf (strcase (strcat (itoa rf) rf)))
(setq r (- 0 r))
(command "UCS" "D" (list lx ly))
(command "UCS" "Z" r)
(command "TEXT" "-1,-0.5" "1" "D" rf)
(command "UCS" "W")
(command "LAYER" "s" "D" "")
(setqvar "orthomode" ort)
(setqvar "cmdecho" cmd)
(setqvar "blipmode" bli)
(setq "error" olderr)
(setq bd_unit nil)
(princ)
);:dpline

```

PROGRAM 2 : IMPOST.LSP

```

(princ "loading IMPOST...")(terpri)
;; utility for superIMPosing STructural forms on digital elevation model
;; 1996. 7.
;;
;;
(defun C:IMPOST (/ s ss cmd bli grd mesh_lyr x1 x2 ms en i entype1
entype2 bdh_lyr txt_ent bdtype bdl bd_hgt roof_type scont cnt crnt ecnt
j k tmp cord min_x max_x min_y max_y px py msh msh_cnt msh_y
mz1 mz2 mz3 mz4 pz lx ly lz cnt_x cnt_y int_cord st_cord end_cord
wall_hgt roof_loc oldx1 oldx2 oldy1 oldy2 rc1 rc2 hst1 hst2 hst3 hst4
hst5 hst6 hst_a hst_b )
;;
;; ----- Error Handling Function 1 -----
(defun clerr (s)
(if (/= "Function cancelled" s)
(princ (strcat "\nError: " s)))
(setqvar "cmdecho" 0)
(setqvar "blipmode" 0)
(setq "error" olderr clerr nil)
(princ))
);:clerr
;; ----- Error Handling Function 2 -----
(defun gerr (ss)
(if (= "bad argument type" ss)
(princ "\nFunction cancelled: object to process not found "))
(setqvar "cmdecho" 0)
(setqvar "blipmode" 0)
(setq "error" olderr gerr nil)
(princ))
);:gerr
;; ----- Main Function -----
(setq olderr "error"
"error" clerr)
(graphscr)
(setq cmd (getvar "cmdecho"))

```

```

(setq bli (getvar "blipmode"))
(setqvar "cmdecho" 0)
(setqvar "blipmode" 0)
(command "UCS" "W")
);:
(setq bdh_lyr "BLDG_FL_" roof_lyr "BLDG_ROOF_"
(setq "error" gerr)
(prompt "\nGrid cells are must be in one layer... "))(terpri)
(setq grd (entget (car (entsel "Select a grid cell : "))))(princ)
(setq mesh_lyr (cdr (assoc 8 grd)))
(setq x1 (cdr (assoc 10 grd)))
(setq x2 (cdr (assoc 11 grd)))
(setq ms (- x2 x1))
(setq en (ssget "X" (list (cons 8 "DIGITLINE_BLDG"))))
(if (/= en nil)
(setq i (sslength en))
(setq gerr nil)
(setq cnt 0 crnt 0)
(setq i (- i 1))
(while (< cnt i)
(setq entype1 (cdr (assoc 0 (entget (ssname en cnt))))))
(setq entype2 (cdr (assoc 0 (entget (ssname en (setq crnt (+ cnt 1)))))))
(if (and (= entype1 "TEXT") (= entype2 "LINE"))
(progn
(setq bdh_lyr "BLDG_FL_" roof_lyr "BLDG_ROOF_"
(setq txt_ent (entget (ssname en cnt)))
(setq bdtype (cdr (assoc 1 txt_ent)))
(command "LAYER" "m" "BLDG_ID_TEXT" "")
(entmod (subst (cons 8 "BLDG_ID_TEXT") (assoc 8 txt_ent) txt_ent))
(setq bdl (strlen bdtype))
(setq bd_hgt (atoi bdtype) roof_type (substr bdtype bdl))
(setq bdh_lyr (strcat bdh_lyr (itoa bd_hgt)
roof_lyr (strcat roof_lyr roof_type))
(command "LAYER" "m" bdh_lyr "")
(command "LAYER" "m" roof_lyr "")
(setq scont (+ cnt 1) crnt (+ cnt 1))
);:if
(if (and (= entype1 "LINE") (= entype2 "LINE"))
(setq cnt (+ cnt 1))
);:if
(if (or (and (= entype1 "LINE") (= entype2 "TEXT")) (= crnt i))
(progn
(setq ecnt (+ cnt 1) crnt (+ cnt 1))
(setq j (- ecnt scont) k 0)
(repeat j
(setq tmp (entget (ssname en scont)) k (+ k 1))
(setq tmp (subst (cons 8 bdh_lyr) (assoc 8 tmp) tmp))
(entmod tmp)
(setq cord (list (car (cdr (assoc 10 tmp)))
(cadr (cdr (assoc 10 tmp))))))
(setq px (car cord) py (cadr cord))
(if (= k 1) (setq min_x px max_x px min_y py max_y py vtx1 cord))
(if (= k 2) (setq vtx2 cord))
(if (= k 3) (setq vtx3 cord))
(if (= k 4) (setq vtx4 cord))
(if (< px min_x) (setq min_x px))
(if (> px max_x) (setq max_x px))
(if (< py min_y) (setq min_y py))
(if (> py max_y) (setq max_y py))
(setq scont (+ scont 1))
);:repeat j
(setq scont (- scont j) k 0)
(repeat j ----- (1)
(setq tmp (entget (ssname en scont)) k (+ k 1))
(setq cord (list (car (cdr (assoc 10 tmp)))
(cadr (cdr (assoc 10 tmp))))))
(setq px (car cord) py (cadr cord))
(setq mx (* (fix (/ px ms)) ms))
(setq my (* (fix (/ py ms)) ms))
(setq msh (ssget "X" (list (cons 8 mesh_lyr)
(cons -4 "=") (list 10 mx my))))
(setq msh_cnt 0)
(setq msh_y nil)
(while (/= msh_y my)
(setq msh_y (cdr (cdr (assoc 10 (entget (ssname msh msh_cnt))))))
(setq msh_cnt (+ msh_cnt 1))
);:while
(setq msh_cnt (- msh_cnt 1))
(setq mz1 (caddr (assoc 10 (entget (ssname msh msh_cnt))))))
(setq mz2 (caddr (assoc 11 (entget (ssname msh msh_cnt))))))
(setq mz3 (caddr (assoc 12 (entget (ssname msh msh_cnt))))))
(setq mz4 (caddr (assoc 13 (entget (ssname msh msh_cnt))))))

```

```

(if (= (- px mx) (- py my))
  (setq pz (+ mz1 (* (/ (distance cord (list mx my))
                        (sqrt (* (expt ms 2) 2))
                        (- mz3 mz1))))))
);:if
(if (< (- px mx) (- py my))
  (progn
    (setq ix (/ (* (- px mx) ms) (- py my)))
    (setq iz (+ mz4 (* (/ ix ms) (- mz3 mz4))))
    (setq pz (+ mz1 (* (/ (distance cord (list mx my))
                        (distance (list (+ mx ix) (+ my ms)) (list mx my)))) (- iz mz1))))))
);:if
(if (> (- px mx) (- py my))
  (progn
    (setq iy (/ (* (- py my) ms) (- px mx)))
    (setq iz (+ mz2 (* (/ iy ms) (- mz3 mz2))))
    (setq pz (+ mz1 (* (/ (distance cord (list mx my))
                        (distance (list (+ mx ms) (+ my iy)) (list mx my)))) (- iz mz1))))))
);:if
(if (= k 1) (setq minz pz))
(if (<= pz minz) (setq minz pz))
(setq scont (+ scont 1))
);:repeat j
(setq scont (- scont j))
(setq min_x (* (fix (/ min_x ms)) ms)
  max_x (* (+ (fix (/ max_x ms)) 1) ms))
(setq min_y (* (fix (/ min_y ms)) ms)
  max_y (* (+ (fix (/ max_y ms)) 1) ms))
(setq cnt_x (fix (+ (/ (- max_x min_x) ms) 1))
  cnt_y (fix (+ (/ (- max_y min_y) ms) 1)))
(repeat j
  (setq tmp (entget (ssname en scont)))
  (setq st_cord (list (car (cdr (assoc 10 tmp)))
    (cadr (cdr (assoc 10 tmp))))))
  (setq end_cord (list (car (cdr (assoc 11 tmp)))
    (cadr (cdr (assoc 11 tmp))))))
  (repeat cnt_x
    (setq int_cord (inters st_cord end_cord
      (list min_x min_y) (list min_x max_y)))
    (if (/= int_cord nil)
      (progn
        (setq mx (car int_cord) my (* (fix (/ (cadr int_cord) ms)) ms))
        (setq msh (ssget "X" (list (cons 8 mesh_lyr)
          (cons -4 "=") (list 10 mx my))))
        (setq msh_cnt 0)
        (setq msh_y nil)
        (while (/= msh_y my)
          (setq msh_y (cadr (cdr (assoc 10
            (entget (ssname msh msh_cnt))))))
            (setq msh_cnt (+ msh_cnt 1))))
        );:while
        (setq msh_cnt (- msh_cnt 1))
        (setq mz1 (caddr (assoc 10 (entget (ssname msh msh_cnt))))))
        (setq mz4 (caddr (assoc 13 (entget (ssname msh msh_cnt))))))
        (setq pz
          (+ mz1 (+ (* (/ (- (cadr int_cord) my) ms) (- mz4 mz1))))))
        (if (<= pz minz) (setq minz pz))
        );:if
        (setq min_x (+ min_x ms))
        );:repeat cnt_x
      (setq min_x (- min_x (* cnt_x ms)))
      (repeat cnt_y
        (setq int_cord (inters st_cord end_cord
          (list min_x min_y) (list max_x min_y )))
        (if (/= int_cord nil)
          (progn
            (setq mx (* (fix (/ (car int_cord) ms)) ms) my (cadr int_cord))
            (setq msh (ssget "X" (list (cons 8 mesh_lyr)
              (cons -4 "=") (list 10 mx my))))
            (setq msh_cnt 0)
            (setq msh_y nil)
            (while (/= msh_y my)
              (setq msh_y (cadr (cdr (assoc 10
                (entget (ssname msh msh_cnt))))))
                (setq msh_cnt (+ msh_cnt 1))))
            );:while
            (setq msh_cnt (- msh_cnt 1))
            (setq mz1 (caddr (assoc 10 (entget (ssname msh msh_cnt))))))
            (setq mz2 (caddr (assoc 11 (entget (ssname msh msh_cnt))))))
            (setq pz
              (+ mz1 (+ (* (/ (- (car int_cord) mx) ms) (- mz2 mz1))))))
            (if (<= pz minz) (setq minz pz))
            );:if
          );:repeat cnt_y
        );:repeat cnt_y
      );:repeat j
    (setq min_y (* (fix (/ min_y ms)) ms)
      max_y (* (+ (fix (/ max_y ms)) 1) ms))
    (setq oldx1 (car (cdr (assoc 10 tmp)))
      oldy1 (cadr (cdr (assoc 10 tmp))))
    (setq oldx2 (car (cdr (assoc 11 tmp)))
      oldy2 (cadr (cdr (assoc 11 tmp))))
    (setq newloc1 (list oldx1 oldy1 minz)
      newloc2 (list oldx2 oldy2 minz))
    (setq tmp (subst (cons 10 newloc1) (assoc 10 tmp) tmp))
    (setq tmp (subst (cons 11 newloc2) (assoc 11 tmp) tmp))
    (setq tmp (append tmp (list (cons 39 wall_hgt))))
    (entmod tmp)
    (setq scont (+ scont 1))
    (if (= roof_type "A")
      (entmake (list (cons 0 "VERTEX") (list 10 oldx1 oldy1 roof_loc))))
    );:repeat j
  (if (= roof_type "A")
    (progn (entmake (list (cons 0 "SEQEND"))
      (solmesh (ssget "L"))))
    (if (or (= roof_type "B") (= roof_type "C") (= roof_type "D"))
      (progn
        (if (> (distance vbx1 vbx2) (distance vbx1 vbx4))
          (progn
            (setq rc1 vbx2 rc2 vbx4)
            (setq vbx2 rc2 vbx4 rc1)
            );:if
          (setq hst1 (list (/ (+ (car vbx1) (car vbx2)) 2)
            (/ (+ (cadr vbx1) (cadr vbx2)) 2)))
            (setq hst2 (list (/ (+ (car vbx3) (car vbx4)) 2)
              (/ (+ (cadr vbx3) (cadr vbx4)) 2)))
            (setq hst3 (list (/ (+ (car hst1) (car hst2)) 2)
              (/ (+ (cadr hst1) (cadr hst2)) 2)))
            (setq hst4 hst3)
            (setq hst5 (list (/ (+ (car hst1) (car hst3)) 2)
              (/ (+ (cadr hst1) (cadr hst3)) 2)))
            (setq hst6 (list (/ (+ (car hst2) (car hst4)) 2)
              (/ (+ (cadr hst2) (cadr hst4)) 2)))
            (if (= roof_type "B")
              (setq lyr_a bdn_lyr lyr_b roof_lyr hst_a hst1 hst_b hst2))
              (if (= roof_type "C")
                (setq lyr_a roof_lyr lyr_b roof_lyr hst_a hst5 hst_b hst6))
                (if (= roof_type "D")
                  (setq lyr_a roof_lyr lyr_b roof_lyr hst_a hst3 hst_b hst4))
                  (entmake (list (cons 0 "3DFACE") (cons 8 lyr_a)
                    (list 10 (car vbx1) (cadr vbx1) roof_loc)
                    (list 11 (car hst_a) (cadr hst_a) (+ roof_loc 2))
                    (list 12 (car vbx2) (cadr vbx2) roof_loc)
                    (list 13 (car vbx2) (cadr vbx2) roof_loc)))
                    (entmake (list (cons 0 "3DFACE") (cons 8 lyr_a)
                    (list 10 (car vbx3) (cadr vbx3) roof_loc)
                    (list 11 (car hst_b) (cadr hst_b) (+ roof_loc 2))
                    (list 12 (car vbx4) (cadr vbx4) roof_loc)
                    (list 13 (car vbx4) (cadr vbx4) roof_loc)))
                    (entmake (list (cons 0 "3DFACE") (cons 8 lyr_b)
                    (list 10 (car vbx1) (cadr vbx1) roof_loc)
                    (list 11 (car hst_a) (cadr hst_a) (+ roof_loc 2))
                    (list 12 (car hst_b) (cadr hst_b) (+ roof_loc 2))
                    (list 13 (car vbx4) (cadr vbx4) roof_loc)))
                    (entmake (list (cons 0 "3DFACE") (cons 8 lyr_b)
                    (list 10 (car vbx2) (cadr vbx2) roof_loc)
                    (list 11 (car hst_a) (cadr hst_a) (+ roof_loc 2))
                    (list 12 (car hst_b) (cadr hst_b) (+ roof_loc 2))
                    (list 13 (car vbx3) (cadr vbx3) roof_loc)))
                    );:if
                    );:progn
                    );:if
                    );:while(if (/= en nil) (prompt "\nprocessing completed..."))
                    (setq "error" olderr clrr nil gerr nil)
                    (setvar "cmdecho" cmd)
                    (setvar "blipmode" bli)
                    (princ)
                    );:impost

```