프랙탈 기법에 의한 울릉도 형상화 사례 연구*

Simulating Uleung Island By The Statistical Fractals

노 용 덕**
Noh Yong Deok

Abstract

In 3D computer graphics, fractal techniques have been applied to terrain models. Even though fractal models have become popular for recreating a wide variety of the shapes found in nature, a specific 3D terrain model such as Uleung Island could not be formulated by statistical fractals easily owing to the random effects. However, by locating the midpoints on the edges and the surface of a specific terrain such as Uleung Island, a similar shape of the terrain model can be simulated. This paper shows the way of simulating 3D Uleung Island terrain model by the statistical fractals wherein the subdivision algorithm is used.

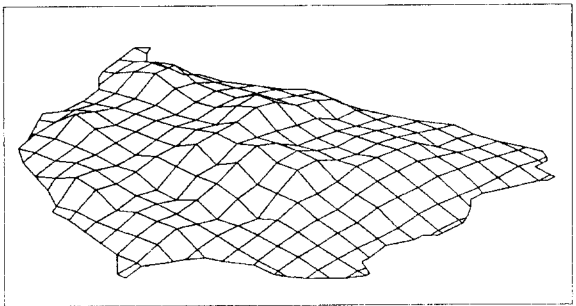
1. 서 론

컴퓨터 그래픽은 매우 다양한 분야에서 사용되고 있으며, 그 중에서도 시뮬레이션 분야에서 컴퓨터 그래픽이 사용된 것은 매우 오래전 부터의 일이다. 그 대표적인 예가 훈련에 응용하기 위하여 컴퓨터 그래픽을 사용하는 것으로, 선박항해사나 자동차 운전사 또는 비행기 조종사를 위한 모의훈련장치가 있다. 모의훈련장치내에서 훈련받는 조종사에게 보여지는 여러 장면들은 다양한 상황을 나타내야 하므로 컴퓨터 그래픽으로 처리한다. 이러한 장면들에서 보이는 배경, 즉, 산이나 강과 같은 것이 어우러져 있는 풍경은 엄밀히 말해서 사실적이지는 않다. 이것은 배경을 실제와 같이 자세한 묘사를 하는데에는 매우 많은 데이타와 엄청난 컴퓨터의 처리 능력을 필요로 하기 때문이다.

컴퓨터 그래픽 분야에서 3 차원 물체를 효과적으로 표현하는 방법에 대한 연구는 계속되어 왔으며, 단순하면서도 효과적인 방법중의 하나가 하나의 대상 물체를 다각형의 형태로 표현하는 것이다. 이러한 방법에 의한 물체의 표현은 많은 데이타를 필요로 하기는 하지만, 어떤 특정한 형태의 물체, 예를 들어서 자동차나 빌딩, 비행기와 같은 물체를 모델화하는데 매우 유효한 방법이다. 이 논문에서 시도하고자 하는 울릉도에 대한 형상화 작업도 3 차원 데이타를 준비하여 표현할 수가 있다. 〈그림 1〉은 다각형의 형태로 데이타를 준비한 후에 그린 울릉도의 모습으로 동남방에서 30도의 눈 위치에서 보이는 모습을 그린

* 이 연구는 대양학원의 연구비 지원에 의하여 이루어진 것임.
** 세종대학교 전산과학과

것이다. 이러한 울릉도 모델을 구하기 위하여 여기서는



〈그림 1〉 Uleung Island

263 개의 꼭지점 데이타와 이 데이타를 이용한 440 개의 면을 필요로 하였다. 물론, 여기서도 보다 자세한 모습의 울릉도를 표현하자면 보다 많은 데이타를 준비하면 된다.

그러나 이와같은 다각형 형태의 데이타에 의한 물체의 표현이나 모델화는 단순하고 효과적이기는 해도, 실제로 대상물체 표현을 위한 데이타를 준비하는 작업은 매우 지루하면서도 어려운 일이다. 또한 데이타의 수에 따라서 물체를 모델화하는 세밀함의 정도가 결정되므로 어느 정도의 데이타를 준비하는 것이 좋은지를 결정하기가 쉽지 않다. 이는 자세한 표현을 요구하면 할수록 많은 데이타가 필요하며, 데이타가 많으면 많을수록 이를 처리하는 컴퓨터의 작업시간이 늘어나기 때문이다. 따라서 이러한 방법은 데이타를 쉽게 구할 수 있는 물체, 예를 들어서 유리잔이나, 구와 같이 어떤 수식과 같은 것에 의하여 규칙적으로 데이타가 만들어지는 물체를 주 대상으로 한다. 데이타가 주어진 식에 의하여 생성되므로 원하는 양의 데이타도 손쉽게 구할 수가 있다.

그러나 컴퓨터 그래픽의 대상이 되는 많은 물체중에는 지형이나 산, 구름, 나무, 숲과 같은 것들은 규칙적인 형상을 갖지 않으며, 따라서 이를 표현하는 다각형 데이타를 준비하는 것이 쉬운 일이 아니다. 이것이 바로 비행 시뮬레이션과 같은 시스템에서 보이는 배경의 그림에서 나무나 산들이 사실적으로 보이지 않고 단순히 나무나 산이라고 느낄 정도로만 표현하는 이유이기도 하다.

최근에 이러한 비규칙적인 성질을 갖는 물체에 대한 모델화 작업이 Mandelbrot[3] 에 의한 프랙탈 기법에 의하여 수행되어 왔다. Mandelbrot 은 분열도형(fractal geometry)이라는 용어를 사용하여 해안선과 같은 자연적인 형상에 대한 성질을 묘사하였다. Mandelbrot 에 의하여 시작된 프랙탈 기법은 2 차원 및 3 차원 지형 형상화 작업에 적용되는 것은 물론 자연에서 볼 수 있는 많은 물체들을 컴퓨터상에서 재창조하고 묘사하는데 매우 효과적으로 사용되고 있다[5].

자연에서 관찰할 수 있는 물체들을 표현하기 위한 다양한 프랙탈 기법이 계속적으로 개발되기는 하였으나, 하나의 방법으로 모든 자연적인 대상 물체를 표현할 수 있는 것은 아니다. 비규칙적인 성질을 갖는 지형을 모델화하는데도 여러가지 프랙탈 기법이 만들어졌는데 일반적으로 크게 나누어 다음과 같은 다섯가지 방식이 있다[8]. 즉, (1) Poisson faulting, (2) Fourier filtering, (3) Midpoint displacement, (4) Successive random addition, (5) Summing band-limited noises 등이 바로 그것으로, 각각의 방식마다 다양한 기법이 개발되었다.

이 논문에서는 세번째 방식에 의한 고정점을 허용하는 프랙탈 기법을 사용하여 울릉도를 형상화하고자 하며, 논문의 순서는 다음과 같다. 먼저 중간점 이동방식(Midpoint displacement)에 따라 개발된 기법들에 대한 조사 및 평가를 보이고, 다음에 여기서 사용되는 고정점을 허용하는 기법의 특징과 알고리즘을 소개하고, 마지막으로 울릉도의 특성을 보이는 데이타의 선택방식과 그에 따른 컴퓨터의 처리 결과와 이 기법의 사용한계에 대한 평가를 결론에서 보인다. 특히, 사용하는 알고리즘 자체에 대한 자세한 분석보다는 울릉도와 같은 지정된 지형에 대한 사례연구에 중점을 둔다.

2. 중간점 이동에 관련된 기법들

중간점 이동 (midpoint displacement) 알고리즘은 Fournier [1,2] 에 의하여 처음으로 개발된 것으로 연속분할(subdivision) 알고리즘이라고도 한다. 이 기법은 하나의 선분을 계속적으로 분할하여 비규칙적인, 예를 들어서 하나의 해안선과 같은 형태를 만들어낸다. Fournier[2] 는 그의 논문에서 하나의 예로 호주 대륙을 형상화 하는 작업을 2 차원으로 보여 주고 있다.

Fournier 의 알고리즘을 기본으로 해서 여러가지 중간점 이동 알고리즘에 관한 기법이 개발되었다. 와이어 프레임 연속분할 방식은 삼각형을 사용하는 연속분할방식을

기본으로 두 개의 점을 사용한 보간법을 적용하여 중간점을 계산한다[2]. 반면에 타일(tile midpoint displacement) 방식은 비선형상에 위치한 3 개이상의 점을 사용한 보간법을 사용하고 있다[10]. 이 기법들은 비교적 효율적이며 구현하기에 좋은 장점이 있는 반면에 계속적으로 분할하는 작업중에 공백이나 비정형적인 결과를 얻을 수 있는 문제가 있다.

Lewis[9] 에 의하여 개발된 일반화된 확률적 연속분할(generalized stochastic subdivision) 방식은 여러 점에 대한 상관관계함수를 사용하고 보간법을 적용하여 중간점을 구하는 기법이다. 즉, 분할시마다 구하는 중간점을 독립적으로 계산하는 것이 아니라 여기에 상관관계가 있는 가우스 값을 더하여 중간점을 구한다. 이 방식은 그 결과에 있어서 인공적이라는 느낌을 다소 경감시키기는 하지만, 삼각형 전체에 있어서 통계적으로 비정형적인 결과를 갖기에 다른 방법과 크게 다른 것은 아니다. Miller[10] 는 비내포성 연속분할(unnested subdivision) 방식도 제안하였는데, 여기서는 중간점을 계산할 적에 이에 관계된 점을 사용하지 않고 구한다는 의미로 쓰인다. 따라서 이는 엄밀히 말하여 중간점 이동방식은 아니다. Lewis 의 방식은 다양하게 적용되는 이점이 있기는 하지만 구현하거나 사용하기에는 매우 불편하고, Miller 의 기법은 구현하기는 쉬워도 공백이 만들어지는 문제가 있다.

Musgrave[8] 는 "noise technique" 라는 기법을 제안하였는데, 지형의 한 부분에서 독립적으로 제어할 수 있는 기능을 제시하고 있다. 그외에도 Szeliski and Terzopoulos [11] 가 제안한 "constrained fractals" 기법에서는 프랙탈 기법과 스프라인 기법을 혼용하는 방식을 사용하고 있다. 여기서는 확률적인 프랙탈의 성질과 결정적인 스프라인의 성질을 적용하여 지형형상화를 시도하고 있다. 이 방식은 연속분할시에 발생하는 제어하기 어려운 점들을 경감시키는 이점을 갖고 있다.

위에서 언급한 기법들은 대개 초기의 삼각형의 데이타에서 어느 정도의 제어권을 갖고서 연속적인 분할을 수행하여 가는 방식이다. 이 기법에 의한 최종적인 물체의 형상화 결과는 항상 초기의 삼각형 경계선을 기준으로 비정형적으로 움직이는 결과를 갖고 오며, 시각적으로 인공적인 대상이 만들어졌다는 느낌을 주는 문제가 있다. 따라서 이러한 기법들은 어떤 특정 대상을 묘사하는데는 부적절하며, 어느 정도 임의의 형태를 갖는 대상을 형상화 하는데 적합하다. 여기서는 연속분할 기법을 사용하되, 중간점을 고정시켜서 원하는 대상 물체의 외형적인 특성을 갖도록 한 후에 랜덤효과를 추후에 허용하는 방식을 사용한다. 이 방식은 연속분할 알고리즘이 재귀적 호출 방식에 의하여 프로그래밍되기에 재귀호출의 매단계마다 고정된 중간점을 데이타로 제공하여야 하며, 단계가 늘어날 때마다 준비해야 할 데이타의 수가 기하급수적으로 늘어나는 문제가 있다. 따라서 원하는 연속분할 단계까지만 고정점을 제공하며, 그 후의 연속분할 단계부터는 랜덤효과를 허용하여 최종적인 지형 데이타를 생성한다.

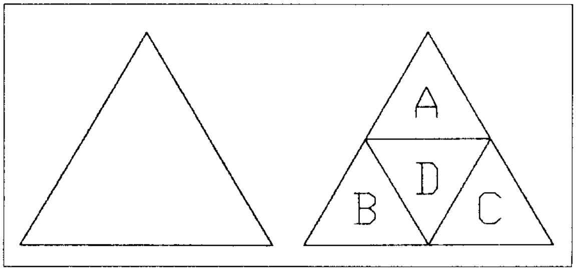
3. 고정점을 사용하는 알고리즘

프랙탈 기법에서의 연속분할방식은 하나의 면을 연속적으로 계속하여 분할하여 새로운 점을 찾아내고 이 새로운 점에 의하여 만들어지는 새로운 면을 기준으로 다시금 분할하는 기법이다. 이 과정을 통하여 하나의 면에 대하여 수없이 많은 작은 면을 만들어 낼 수가 있다. 이러한 과정은 이론적으로 무한히 계속 될 수가 있으나, 대개는 원하는 횟수만큼의 유한한 반복만을 허용하거나 원하는 대상 물체묘사의 세밀함의 정도에 따라서, 또는 컴퓨터의 성능이나 출력장치의 해상도에 의존하여 반복회수를 정하기도 한다.

이때 사용되는 면은 대개 삼각형을 사용하며, 몇몇 사람들[12]에 의하여 사각형이나 육각형에 대하여서도 시도되었다. 만들어지는 새로운 삼각형의 모양은 중간점이 위치하는 장소에 따라서 조금씩 원래의 모양과 달라지게 되며, 중간점은 일반적으로 가우스 난수에 의하여 결정하기에, 이때 생기는 랜덤효과를 이용하여 지형을 형상화한다.

초기의 삼각형에서 지정한 3 개의 꼭지점의 위치는 연속분할작업이 진행되어도 값이 바뀌지 않는다. 그러나 이 점을 기본으로 하여 만들어지는 중간점들은 랜덤효과에 의하여 값을 임의로 결정한다. 따라서 연속분할이 진행됨에 따라서 생기는 결과의 삼각형 모습을 예견하기는 어려운 일이다.

초기의 삼각형에서 한번 분할작업을 수행하되 랜덤효과를 고려하지 않고 만들어진 새로운 삼각형이 〈그림 2〉에 보인다. 이러한 그림을 그리기 위하여는 A,B,C, 3 개의 삼각형만 있으면 가능하다. 그러나 이때 만들어진 데이타를 이용하여 색칠을 하거나 빛을 넣거나 하는 작업을 수행하



〈그림 2〉 Triangle Subdivision

고자 하려면 A,B,C,D, 4 개의 삼각형이 필요하게 된다. 이것은 하나의 표면을 구성하는 모든 다각형에 대하여 색이나 빛을 처리하는 작업을 수행하여야 하기 때문이다. 따라서 연속분할작업시에 기억시켜야 할 삼각형의 수는 다음의 식에 의한다.

$$N_t = \sum_{k=1}^{n} K + \sum_{k=1}^{n-1} K \qquad n = 2^i \tag{1}$$

여기서 i 는 연속분할반복 횟수이다.

반복회수가 증가함에 따라서 만들어지는 삼각형의 수와 기억해야할 데이타의 수가 기하급수적으로 늘어난다. 기억해야할 데이타의 수는 삼각형마다 3 개의 데이타가 필요하다고 볼 때에, 식 (1) 에 의하여 N_t*3으로 볼 수 있다. 그러나 하나의 점을 경우에 따라서는 6 개의 삼각형이 공유할 수도 있다. 이때, 중복되어 사용되는 데이타를 하나의 데이타로 간주하면, 연속분할시마다 만들어지는 전체 데이타 수는 다음의 식에 의한다.

$$N_p = 0 \qquad \text{if } i=0 \tag{2}$$
$$= \sum_{k=1}^{n} K \qquad \text{otherwise,} \quad n = (2^i + 1)$$

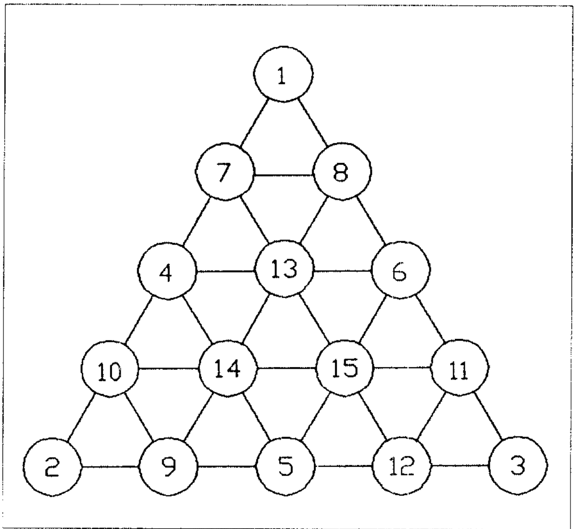
식 (2) 에서 i 는 반복회수, N_p는 꼭지점의 수이다. 식 (2) 에서 보듯이 꼭지점의 수는 3, 6, 15, 45, 153,, 의 순으로 기하급수적으로 늘어난다. 따라서 연속분할시마다 새로이 계산하여야 하는 중간점의 수는 3, 9, 30, 108,, 과 같이 증가하고 있다. 이 중간점들은 일반적으로 랜덤하게 만들어지나, 이의 위치를 대상 물체의 특성을 반영하는 특정지점에 강제로 위치하게 제어함으로써 랜덤효과를 줄이고 대신에 원하는 물체을 표현하는 데이타를 구할 수가 있다.

이를 구현하기 위한 알고리즘이 〈그림 3〉에 나와 있다. 연속분할이 진행함에 따라서 하나의 삼각형이 4 개의 삼각형으로 분할됨으로 재귀호출 단계마다 매번 4 번씩 이루어지고 있다. 이러한 재귀호출로 인하여 중간점을 미리 지정한 값으로 대체하는데 이를 원하는 단계까지만을 허용한다. 〈그림 3〉의 알고리즘에서는 2 단계의 재귀호출까지만이 지정한 데이타 사용을 허용하고 있다.

```
double ax[15],ay[15],az[15];
int       maxiter;
int       stackno = 0;
const     resolution;

procedure subdivision(x,y,z,roughness,iteration:real);
begin
 if(length of line segment < resolution) then iteration=0;
 if(iteration = 0)
  begin
    Move to first point;
    Line to second point;
    Line to third point;
    Line to first point;
  end

 else
  begin
    {switch(maxiter-n) {

      case 0 : insert 4th,5th,6th midpoints; break;
      case 1 :
        {stackno = stackno + 1;
        if ( stackno == 1) insert 7th,8th,13th midpoints;
        if ( stackno == 2) insert 9th,10th,14th midpoints;
        if ( stackno == 3) insert 11th,12th,15th midpoints;
        if ( stackno == 4) insert 13th,14th,15th midpoints;}
        default: compute roughness and mid-points;}

     subdivision(x,y,z,roughness,iteration-1);
     subdivision(x,y,z,roughness,iteration-1);
     subdivision(x,y,z,roughness,iteration-1);
     subdivision(x,y,z,roughness,iteration-1);
 end
end;
```

〈그림 3〉 중간점을 허용하는 연속분할 알고리즘

2 단계에서의 재귀호출 이후의 단계에서는 기존의 연속분할방식에 의하여 데이타를 만들어 나간다. 이때 제공되는 데이타의 위치는 〈그림 3〉의 알고리즘의 경우, 〈그림 4〉와 같은 순서로 지정되어야 한다. 그러나 초기에 제공



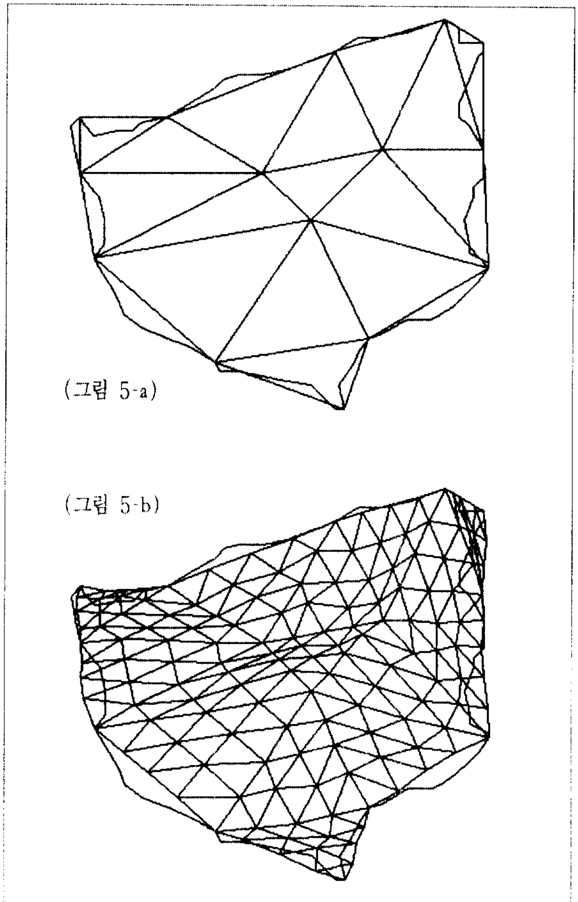
〈그림 4〉 The order of midpoints

하는 데이타가 특정지형의 특성을 반영하는 데이타이기에 그 후에 만들어지는 중간점들이 랜덤하게 생성된다고 해도 특정 지형의 외형적인 특성이 어느 정도는 그대로 남는다. 따라서 이러한 방식에 의하여 우리는 비규칙적인 형태를 갖는 특정 대상의 외형적인 특성을 어느 정도 유지하면서도 그에 필요한 원하는 만큼의 다각형 데이타를 만들어 낼 수가 있다. 즉, 연속분할방식에서 이용하는 랜덤효과를 허용하기는 해도 전체적인 외양은 그대로 유지할 수가 있으며, 원하는 만큼의 데이타를 최소한의 기본 데이타를 제공하여 얻을 수가 있다.

4. 울릉도 형상화 작업

〈그림 3〉의 알고리즘을 이용하여 울릉도에 대한 형상화 작업을 시도하였다. 울릉도는 한반도의 동쪽에 위치한 작은 섬으로, 가운데에 984 미터 높이의 성인봉을 비롯한 여러개의 봉우리가 있고 지표면은 매우 울퉁불퉁한 특징을 갖고 있다.

프랙탈을 이용한 지형 데이타를 생성하기 위하여 먼저 3 개의 꼭지점을 선택한다. 이 점은 해안선의 점으로 가능한 균등한 거리를 둔다. 이를 근거로 프랙탈 기법을 적용하면, 프랙탈에서의 랜덤효과로 인하여 울릉도와는 전혀 다른 모습의 지형이 만들어진다. 여기에서 랜덤효과를

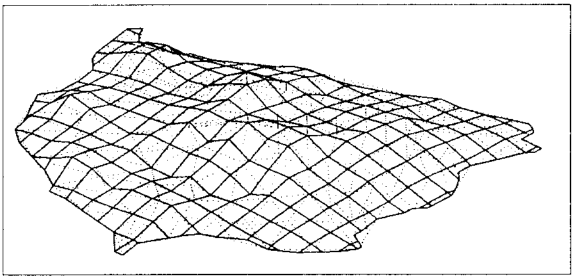


〈그림 5〉 Uleung Island and 15 Midpoints

구하는 값(roughness)을 여러가지로 주어도 원하는 형태의 울릉도 모습이 구하여지지 않는다. 이는 울릉도라는 특수한 형상을 만드는 랜덤효과를 얻기에는 지정된 꼭지점 데이타의 수가 너무 적기때문이다.

따라서 위의 3 개의 점을 포함한 15 개의 점을 울릉도의 지표면상의 점에서 선택하여 위의 알고리즘을 적용한다. 〈그림 5-a〉는 선택한 15개의 점의 위치를 보여주고 있다. 〈그림 5-a〉에서 보듯이 15개의 점이 어느 정도의 울릉도의 외형적인 특성을 보여주고 있다. 또한 섬 내부에도 꼭지점이 위치하게 되므로, 성인봉과 같은 봉우리의 높이나 다른 지표면의 높이를 데이타로 지정할 수가 있어서 울릉도 지표면의 특성을 어느 정도 반영할 수가 있게 된다. 그러나 울릉도는 상대적으로 거칠은 해안선과 고르지

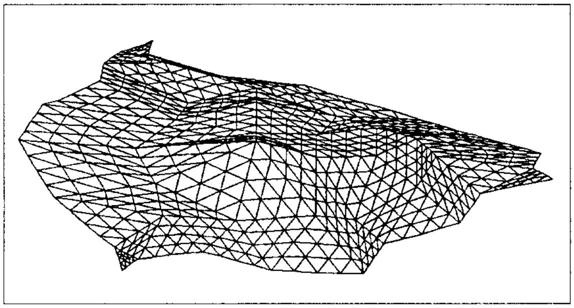
못한 지표면을 갖고 있어서 15 개의 점으로는 울릉도와 유사한 형상을 얻고 실제의 모습과의 차이점을 많이 줄이기는 하였어도 만족스럽지는 못하다. 〈그림 5-b〉는 15개의 고정점을 허용한 프랙탈 기법을 사용하여 만든 울릉도의 모습으로 4 번 연속분할 작업에 의하여 얻어진 데이타로 그린 울릉도이다.



〈그림 6〉 Uleung Island drawn by fractals

재귀호출에서의 제 3 단계까지 데이타 지정을 허용하는 경우는 모두 45 개의 데이타를 필요로 한다. 이 경우는 〈그림 3〉 알고리즘의 switch 문에 'case 2 :' 문장을 새로이 삽입하여 줌으로써 처리할 수가 있다. 〈그림 6〉은 45 개의 데이타를 준비하여 울릉도를 형상화한 것이다. 이때, 랜덤효과를 주기위한 값으로 -1 을 사용하였고, 4 단계까지의 재귀호출을 허용하였다. 45개의 데이타는 되도록 울릉도의 특성을 나타내기 위하여 섬에 있는 봉우리와 골짜기의 위치에 해당하는 지표면의 값을 선택하였다. 〈그림 6〉에서 직선에 의한 울릉도의 모습은 〈그림 1〉의 데이타에 의한 그림이고 점선에 의한 울릉도는 프랙탈에 의하여 생성된 데이타를 사용하여 그린 것이다. 〈그림 1〉과 마찬가지로 동남방에서 30 도의 눈높이에서 그린 결과이다. 여기에서 보듯이 〈그림 1〉에 의한 울릉도와 프랙탈에 의한 그림의 차이가 그다지 크지 않음을 알 수가 있다. 지표면이나 해안선의 굴곡은 랜덤효과 값을 적당히 조절하여 주면 되며, 이는 반복작업에 의하여 원하는 값을 선택한다.

참고로 〈그림 7〉은 같은 데이타로 제 5 단계까지의 재귀호출을 허용하여 울릉도를 그린 것이다. 여기서 해안선이나 지표면을 보다 거칠게 하거나 굴곡을 많이 주고자 하는 경우에는 재귀호출의 단계를 보다 많이 허용하고 랜덤효과의 값을 보다 크거나 작게 한다. 결국, 〈그림 7〉은 단지 45개의 꼭지점 데이타를 제공하여 울릉도의 특성을



〈그림 7〉 Uleung Island with N=5

나타내는 561 개의 꼭지점과 이 꼭지점에 의하여 정의된 1024개의 면을 구성하는 데이타가 자동적으로 생성된 결과에 따라서 그려진 것이다.

5. 요약 및 결론

하나의 물체를 컴퓨터상에서 형상화하기 위한 많은 기법이 개발되어 왔다. 컴퓨터 그래픽에서는 매우 자세한 묘사를 필요로하는 물체가 있는 반면에 대략적인 윤곽으로 원하는 물체가 있어서 그 특성에 따라서 여러가지 다양한 기법이 개발되어 왔다. 정확한 데이타를 요하는 경우는 일반적인 다각형 형태의 그물형 데이타를 준비하지만, 그렇지 않은 자연현상에서의 물체, 즉, 산이나 구름, 숲, 나무와 같은 경우는 프랙탈 기법을 사용할 수 있다. 본 논문에서는 고정된 중간점을 허용하는 프랙탈 기법을 이용하여 울릉도의 특성을 유지하면서 울릉도를 그리는데 필요한 3 차원 데이타를 생성하는 과정을 보이고 거기서 나온 결과를 보였다.

이 방식에서는 원하는 울릉도의 모양을 구하기까지 여러가지의 랜덤요소의 값을 주고 시도를 하여야 하며 초기의 기준값을 어떻게 주어야 하는지가 불분명한 단점이 있다. 이것은 고정된 기준값을 단순히 대상물체의 특성이 될만한 점을 단순히 지적하여 주어야만 하기 때문이다. 현재로는 이에 대한 정확하고도 효과적인 방법은 없는 것처럼 보인다. 그러나 이 방식에서는 매우 적은 데이타만을 준비하고도 어느 정도 대상물체의 특성을 반영하는 많은 데이타를 원하는 만큼 얻을 수가 있는 장점이 있어서, 비행시뮬레이션에서의 배경 그림의 묘사나, 영화, 특수효과를 만드는 작업등에서 효과적으로 사용될 수가 있을 것이

다.

REFERENCES

[1] Fournier, A. and Reeves W.T., "A Simple Model of Ocean Waves", Computer Graphics, Vol.20, No.4, pp. 75-84

[2] Fournier, A. and Fussell D. and Carpenter L., "Computer rendering of stochastic models", Communications of the ACM, Vol.25, No.6, pp.371-84

[3] Hearn,D., Computer Graphics, pp.205-210, Prentice-Hall, 1987

[4] Mandelbrot, B.B., "Comment on Computer Rendering of Fractal Stochastic Models", Communications of the ACM, Vol.25, No.8, pp.581-584, 1982.

[5] Mandelbrot, B.B., "The Fractal Geometry of Nature", W.H.Freeman, San Francisco, 1982

[6] Perlin, K., "An Image Synthesizer", Computer Graphics, Vol.19, No.3, pp.287-296

[7] Watt,L., Fundamentals of Three-Dimensional Computer Graphics, pp.255-261, Addison Wesley, 1989.

[8] Musgrave,F.K., Kolb,C.E., and Mace, R.S., "The Synthesis and Rendering of Eroded Fractal Terrains", Computer Graphics, Vol 23, No.3, pp.41-50, July, 1989.

[9] Lewis,J.P., "Generalized Stochastic Subdivision", ACM Transactions on Graphics, Vol.6, No.3, pp.167-190, July, 1987.

[10] Miller, Gavin S.P., "The Definition and Rendering of Terrain Maps", Computer Graphics, Vol.20, No.4, pp. 39-48, 1986.

[11] Szeliski, R., and Demetri Terzopoulos, "From Splines to Fractals", Computer Graphics, Vol.23, No.3, pp. 51-60, July, 1989.

[12] Peitgen, H.O. and Dietmar Saupe (eds.), "The Science of Fractal Images", Springer-Verlag, New York, 1988

● 저자소개 ●

노용덕
1976년 서울대학교 산업공학과 졸업
1984년 AUBURN Univ I.E. 석사
1987년 AUBURN Univ I.E. 박사
1976 - 1981, 1987년 국방과학연구소
1985년 CDP from ICCP
1988 - 현재 세종대학교 전산과학과 부교수
관심분야: 컴퓨터 그래픽스, 컴퓨터 시뮬레이션