

삼각 격자를 이용한 Z-map 모델

박배용* · 안정호**

Z-map Model Using Triangular Grids

Park, Pae-Yong, Ahn, Jeong-Ho

Key Words: Triangular Grid(삼각 격자), Z-map**Abstract**

Prior to the downloading of the NC codes to a machining center, the NC tool-path can be verified in a computer. The Z-map is one of the tools for the verification of NC tool-path. The Z-map is a two dimensional array in which the height values of the Z-axis direction vectors are stored. The Z-axis direction vectors are arranged in a rectangular grid pattern on the XY plane. The accuracy of the simulation comes from the grid interval. In the rectangular Z-map, the distances between the grid points are different. The distance in diagonal direction is larger than those in X or Y axis directions. For the rendering of the Z-map, a rectangular grid is divided into two triangular facets. Depending on the selection of a diagonal, there are two different cases.

In this paper, triangular Z-map, in which the Z-axis direction vectors are arranged in a triangular grid pattern on XY plane, is proposed. In the triangular Z-map, the distances between grid points are equal. There is no ambiguity to make triangular facets for the rendering.

1. 서 론

Z-map은 1978년 Anderson⁽¹⁾에 의해 개발되었고, 그 후 많은 연구자들에 의해 그 종류 및 적용도가 증가되었다. 기존의 일반적인 Z-map의 형태는 3차원 공간의 XY평면을 유한한 수의 사각격자로 나누고, 그 격자 점마다 원래의 모델이 갖는 Z축 좌표 값을 저장하는 것이다.

일반적인 Z-map 데이터는 하나의 격자점에 하나의 Z축 값을 저장하게 된다. 따라서, 5축 CNC가공 등에 있어서 문제점을 갖게 된다. 즉, 공간상에 가운데가 비어있는 부분 등의 가공에 있어서 하나의 격자점에 여러 개의 Z값을 저장해야 하는 문제가 발생한다. 하나의 격자점에 여러 개의 Z값을 저장하는 방법이 Wang⁽²⁾, Van Hook⁽³⁾등에 의해 개발되었다. Chappel⁽⁴⁾은 곡면의 특정점에 법선벡터를 정의하고, 그 법선벡터의 곡면상의 위치와 법선벡터 자체를 저장하는 방법을 개발하였다. 이 방법에서는 NC가공 시 물레이션에서의 과절삭, 미절삭 등이 저장된 법선벡터의 길이만으로 확인될 수 있다. 다른 형

* 숭실대학교 대학원

** 숭실대학교

태로써, 법선벡터 대신 Z축 방향벡터(Height vector)를 사용할 수도 있다.

이러한 Z-map 모델 형식은 X, Y축의 격자 간격에 의해 정밀도가 결정된다. Z-map 모델은 그 데이터 구조가 간단하고, 가공 시뮬레이션시 수학적 계산이 일반적인 곡면이나 솔리드 모델보다 쉽게 적용될 수 있다. 또한, 3차원 렌더링에 의한 대상의 표현에 있어서도 공간상의 평면을 쉽게 정의 할 수 있는 장점을 가진다. 그러나, 사각격자를 이용하는 기존의 방법은 몇 가지 문제를 가진다. 먼저, Z-map 모델을 가시화 하기 위해서는 Z-map 데이터로부터 평면을 구성해야 한다. 이 때, 사각 격자로는 Fig 1에서와 같이 공간상에서 유일한 하나의 평면을 구성하지 못한다. 즉, 주어진 4개의 격자점에 의해 결정되는 평면은 두 개인 것이다. 따라서, 적절한 평면을 찾아내기 위해서는 대각선의 선택에 대한 고려가 필요하게 된다.

또한, Fig 2에서와 같이 하나의 중심 격자점으로부터 둘러싸고 있는 8개의 격자점들에 대한 거리가 서로 다르다. 중심 격자점 주위의 8개의 격자점들이 중심의 한 점에 가하는 가중치가 다르다는 것이다. 평면을 구성하는 하나의 격자점에서 평활화된 법선벡터를 구할 때 등의 계산에 있어서 균일한 가중치를 줄 수 없게 된다.

본 연구에서는 삼차원 공간의 XY평면을 사각 격자로 나누지 않고, 삼각 격자로 나누어 Z-map을 생성하는 방법을 제시한다.

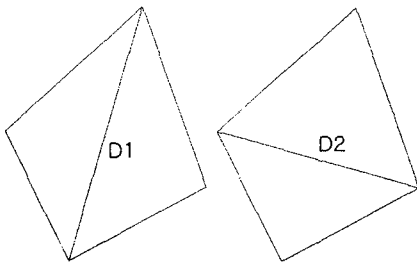


Fig 1. Ambiguity by the selection of a diagonal of rectangular Z-map.

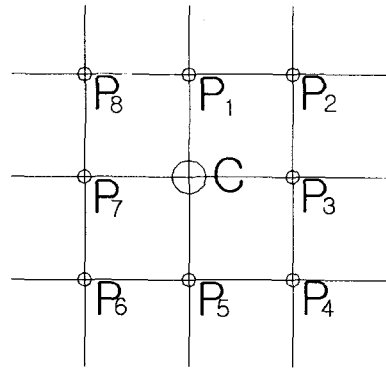


Fig 2. Different distances between a grid point and surrounding points of rectangular Z-map.

2. 삼각 격자 Z-map

본 연구에서 제시하는 삼각 격자인 경우엔, 항상 격자 평면상의 근접한 세 개의 격자점을 이용하여 유일한 하나의 평면을 공간상에 구성할 수 있다. 사각 격자를 이용할 때의 대각선 선택의 문제는 발생하지 않는다. 또한, Fig 3에서와 같이 중심점으로부터, 둘러싸고 있는 주위의 6개 격자점까지의 거리가 일정하다. 따라서, 주위의 6개 격자점들이 중심의 한 점에 균일한 가중치를 줄 수 있게 된다.

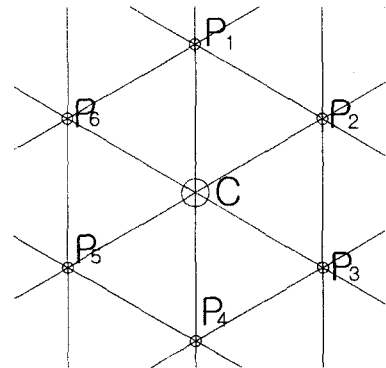


Fig 3. Uniform distances between a grid point and surrounding points of triangular Z-map.

2.1 삼각 격자의 좌표 맵핑(Mapping)

모든 삼각 격자점들은 Fig. 4에서와 같이, S-T 사교 좌표계의 두 기저벡터(Basis vectors) \vec{u} , \vec{v} 의 합으로 표현된다. 따라서, 2차원 배열의 두 첨자들의 순차적인 인덱싱(Indexing)에 의해 삼각 격자점들의 직교 좌표 계에서의 좌표 값을 계산하는 과정이 필요하다. Fig. 4는 \vec{u} , \vec{v} 를 단위벡터로 하였을 때의 맵핑 모습을 보인다. 직교 좌표 계의 한 점 \vec{P}_{xy} 를 구하는 계산 과정은 다음과 같다.

$$\begin{aligned} \vec{P}_{xy} &= s\vec{u} + t\vec{v} \\ &= (u_x + tv_x, su_y + tv_y) \end{aligned} \quad (1)$$

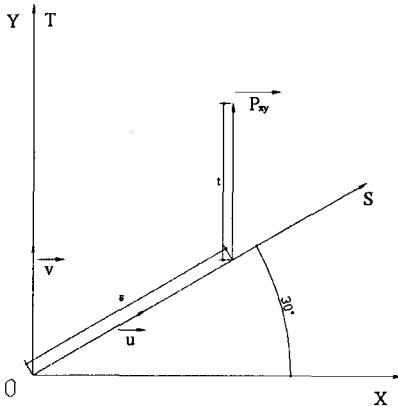


Fig 4. Mapping from S-T oblique coordinate system to X-Y orthogonal coordinate system

식(1)의 \vec{u} , \vec{v} 의 계수(s , t)는 Fig. 5에서와 같이 형성된다. \vec{u} , \vec{v} 의 정수(Integer Number) 계수를 갖는 일반적인 격자점들과, 세로 방향의 경계선 부분에서의 s점, e점 같은 경계 격자점들로 나눌 수 있다. 경계 격자점들은 계산의 효율성을 위해, Z-map데이터를 저장하는 배열에 저장하지 않고 따로 저장하거나, 필요한 때에 임시로 만들어 사용할 수 있다.

일반적인 격자점들에 대한 두 기저벡터의 계수들은 2차원 배열의 두 인덱스(i, j)들에 의해 구해내야 한다. \vec{u} 의 계수 인덱스는 배열의 인덱스

i 를 그대로 사용하면 된다. \vec{v} 의 계수 인덱스는 j 를 그대로 사용할 수 없으므로, i 의 수열에 따른 변화 특성을 고려하여 구한다.

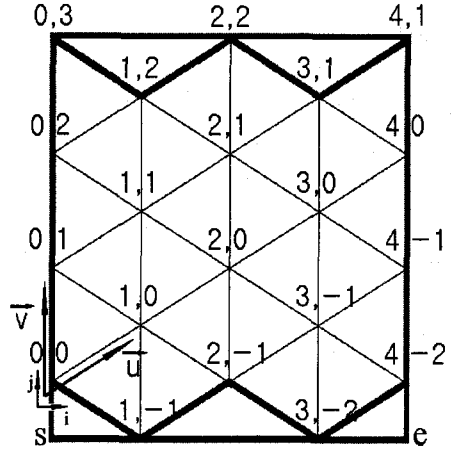


Fig 5. Basis vector (\vec{u} , \vec{v})'s coefficients for coordinate mapping.

배열의 인덱스(i, j)의 변화에 따른 \vec{u} , \vec{v} 의 계수 형성과정을 표로 정리하면 Table 1과 같다.

i \ j	0	1	2	3	4	5	
0	0	-1	-1	-2	-2	-3	...
1	1	0	0	-1	-1	-2	...
2	2	1	1	0	0	-1	...
3	3	2	2	1	1	0	...
...
s	i	i	i	i	i	i	...
t	j	j-1	j-1	j-2	j-2	...	j-k

Table 1. Series of the array indices and basis vector's coefficients of triangular Z-map.

Table 1의 j행은 계차수열의 형태를 지닌다. 아래와 같은 방법으로 i, j 의 배열 인덱스로부터 실제 삼각 격자점들에 있어서 기저벡터들의 계수를 구하여 삼각격자 생성 루틴으로 프로세스를 옮길 수 있다. k 는, i 에 따른 j 의 수열이 계차수열이 됨으로, 그 차에 해당하는 변수다.

```

for j = 0, j < N, j = j+1
  for i = 0, k = 0, i < M, i = i+1
  {
    삼각격자생성루틴(i, j-k)
    if k/2 = 0
      then k = k + 1
    }
  end for
end for

```

삼각격자 생성루틴은 가공 대상 모델과 생성된 삼각격자 평면에서 시작되는 Z축 방향벡터와의 교점계산을 하는 것이다. 계산된 교점의 Z축 성분이 2차원 배열에 저장되면 삼각격자를 이용한 Z-map 모델은 완성된다.

2.2 가시화(Visualization)

삼각 격자 Z-map의 가시화에 사용된 OpenGL은 SiliconGraphics사가 개발한 그래픽 라이브러리다. 이를 이용해 간단한 밀링 가공 흔적을 만들고 렌더링(Rendering) 하였다.

2.2.1 공간상의 평면구성

본 연구에서는 삼각격자를 이용하여 Z-map 데이터를 산출하였으므로, 사각격자에서와 같이 4개의 격자점으로 이루어진 삼각평면 중에서 적절한 삼각형을 찾는 알고리즘은 필요치 않다. 삼각평면을 이루는 격자점들에 대한 인덱스는 사각격자에서와 같이 주어지는 한 점의 Z-map데이터 배열로부터 간단히 구해진다.

2.2.2 법선벡터의 평활화

각각의 삼각평면의 법선벡터를 계산하여 그 법선벡터를 구하여 가시화에 이용하는 것은 간단하다. 그러나, 현재의 삼각평면과 인접한 삼각평면 사이의 기울기의 차이가 현저하거나 빈번히 발생하는 경우엔 가시화에 있어서 그 결과가 좋지 않다. 이러한 문제를 해결키 위해, 삼각평면의 법선벡터를 사용하지 않고, 삼각 격자점에서의 법선벡터를 사용하였다. 즉, 하나의 삼각격자점을 둘러싸고 있는 6개의 삼각평면에서 각각 법선벡

터들을 구하고, 그것들을 평균화 및 정규화 하여 격자점의 법선벡터로 지정하는 것이다. 이렇게 함으로써 법선벡터의 평활화 효과를 얻어, 보다 나은 결과를 얻을 수 있다.

2.3 적용 및 고찰

Fig 6 ~ Fig 9는 Table 1과 같은 시뮬레이션 조건으로 밀링 가공을 묘사한 것이다. 공구는 볼 엔드밀이며, 가공 대상 재료는 직육면체 블록으로 가정하였다. Fig 6~8은 일부분을 보인 것이다.

그림	Fig 6-8	Fig 9
격자점의 수	100×100	200×200
$\Delta x, \Delta y$	4, 4	2.5, 2.5
\vec{u}	(3.464, 2.)	(2.165, 1.25)
\vec{v}	(0, 4.)	(0, 2.5)
공구 반경	60	60
공구 경로간 간격	72	72

Table 1. Simulation conditions for Fig 6~Fig 9

Fig 6은 OpenGL의 삼각형 프리미티브(GL_TRIANGLES)를 사용한 것으로, 법선벡터의 평활화를 시행치 않은 장면이다. Fig 7은 Fig 6과 같은 조건에 법선벡터의 평활화를 시행한 장면이다. Fig 8은 Fig 7의 조건에 스펙큘러(Specular) 조명을 추가한 장면이다. Fig 9는 전체적인 모습을 보인다. 각각의 조건에 따라 결과의 차이가 큼을 알 수 있다.

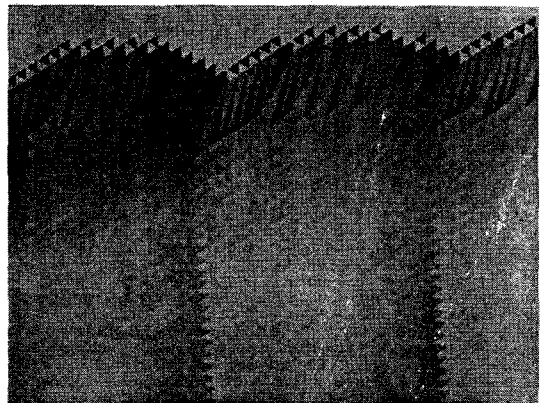


Fig 6. Rendered image of a Z-map model.

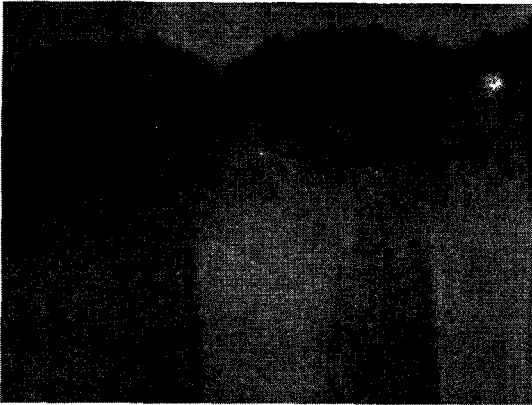


Fig 7. Rendered image with normal vector smoothing.

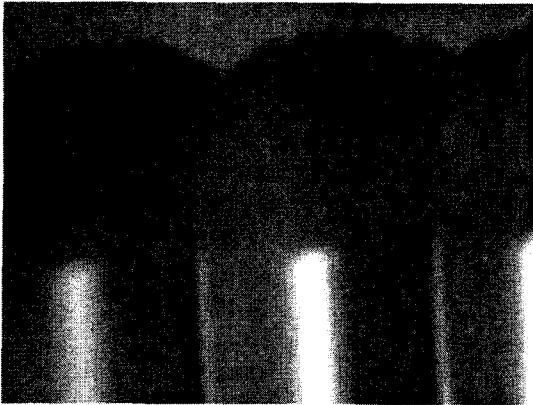


Fig 8. Rendered image with smoothing and specular light source.

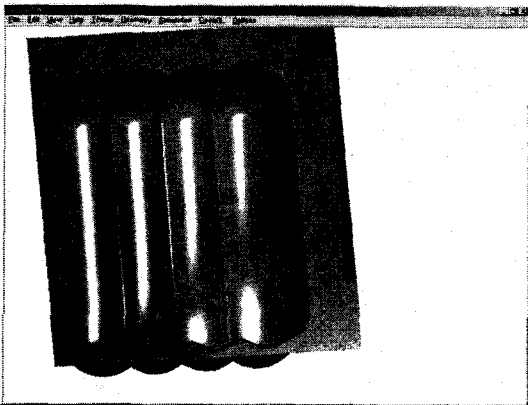


Fig 9. Full rendered image with smoothing and specular light source.

3. 결론

본 연구에서는, 삼각 격자를 이용한 Z-map 모델을 제시하였다. 또한, 간단한 가공 시뮬레이션을 보였고, OpenGL을 이용하여 그 결과를 가시화 할 수 있었다. 앞으로, NC 가공 시뮬레이션에 있어서, 사각 격자를 이용한 Z-map과 삼각 격자를 이용한 Z-map의 수치적인 차이를 비교 및 분석 하고자 한다.

참고문헌

1. Anderson, R.O., 1978, "Detecting and eliminating collisions in NC machining", *Computer-Aided Design*, 10(4), pp. 231-7
2. Wang, W.P. and Wang, K.K., 1986, "Geometric modeling for swept volume of moving solids", *IEEE Computer Graphics and Applications*, Dec., 6(12), pp. 8-17
3. Van Hook, T., 1986, "Real-time shaded NC milling display", *Computer Graphics (Proc. SIGGRAPH)*, Aug., 20(4), pp. 15-20
4. I.T. Chappel, 1983, "The Use of Vectors to Simulate Material Removed by Numerically Controlled Milling", *Computer-Aided Design*, Vol. 15, No. 3, pp. 156-158
5. Jerard, R.B., Drysdale, R.L. and Hauck, K. Schaudt, B. and Magewick, J. (1989) "Methods for detecting errors in sculptured surface machining", *IEEE Computer Graphics and Applications*, 9(1), 26-39
6. B.K. Choi, R.B. Jerard, 1998, *Sculptured Surface Machining Theory and Applications*, Kluwer Academic Publishers, pp. 110-112