

## 3축 NC 가공을 위한 CL Z-map 모델링 방법의 비교 연구

박정환<sup>1</sup> · 정연찬<sup>2</sup> · 최병규<sup>3</sup>

<sup>1</sup>영남대학교 기계공학부 / <sup>2</sup>큐빅기술연구소 / <sup>3</sup>한국과학기술원 산업공학과

### Comparative Study of CL Z-map Modeling for 3-Axis NC Machining

Jung-Wan Park<sup>1</sup> · Yun-Chan Chung<sup>2</sup> · Byoung-Kyu Choi<sup>3</sup>

Gouge-free tool-path generation is an important issue in mold & die machining and researches on cutter interference avoidance can be found in many articles. One of the various methods is construction of tool-offset surface or cutter-location (CL) surface on which the cutter-center point (CL-point) locates. Provided that the CL surface is represented in a suitable form, cutter-interference avoidance can be performed without the burden of computing CL data for every cutter-contact (CC) point. In the paper, various methods of constructing a CL surface in the z-map form are presented, where z-map is a special form of discrete nonparametric representation in which the height values at grid points on the xy-plane are stored as a 2D array  $z[i, j]$ .

#### 1. 서론

일반적으로 금형 형상부 모델은 수십~수천 개의 자유곡면 (sculptured-surface)으로 구성된 복합곡면(compound surface)으로 모델링하고 있다(Park *et al.*, 1999). 대부분의 CAD 시스템은 NURBS (Non Uniform Rational B-Spline) 등의 매개변수형(parametric) 곡면 모델을 기반으로 함으로써 복잡한 형상의 모델링이 가능토록 한다.

가공대상 곡면 모델이 만들어지면, 공정계획을 거쳐 NC 가공을 위한 가공데이터, 즉 CL(cutter-location: 공구기준점) 데이터를 생성한다. 이때 자유곡면 가공경로는 크게 두 가지로 구분할 수 있는데, 매개변수형 곡면의 등경곡선(iso-parametric curve)을 따라 공구가 움직이는 매개변수 방식과  $xy$  평면상에서 정의된 직선을 따라 공구가 이동하는 카테지안(Cartesian) 방식으로 구분할 수 있다. Cartesian 방식은 다시 CC-Cartesian 및 CL-Cartesian으로 구분할 수 있다(Choi and Jerard, 1999). 전자의 경우 가공대상 곡면을 수직평면으로 절단하여 얻어진 개별 CC(cutter-contact: 공구 접촉점) 데이터에 대해 공구간섭을 제거하는 방식을 사용하며(Hwang, 1992; Lee and Jerard, 1994), 후자의 경우는 공구 기준점이 놓이는 공구 오프셋 곡면 즉, 공구기준면(CL-surface)을 계산하여(Choi and Jerard, 1999) 이를 수직평면으로 절단하여 공구경로를 생성하게 된다.

자유곡면의 CL 곡면은 삼각형 등의 다면체(facet) 모델, 또는

z-map 모델(Takeuchi *et al.*, 1989; Saito and Takahashi, 1991; 송수용 & 김석일, 1996) 등 비매개변수형 곡면모델을 이용하는 것이 일반적이다. 특히 z-map 모델의 경우 데이터 구조의 단순함으로 인해 오프셋팅이 단순하고 CL 곡면 구축 후 Cartesian 공구경로 계산(즉, 수직평면과의 교선 계산)이 간단하다는 장점을 가진다(Takeuchi *et al.*, 1989; Choi *et al.*, 1994). 아울러 CL 곡면 계산 과정에서 발생할 수 있는 특이성(singularity) 문제(Do Carmo, 1976)를 비교적 간단하게 해결할 수 있는 장점도 가지고 있다. 이밖에 절삭 모의가공 등 다양한 분야에서 z-map 모델의 적용 예를 찾아볼 수 있다(Choi *et al.*, 1995; 송수용 & 김석일, 1996; 정연찬, 1996).

본 논문에서는 매개변수형 자유곡면으로부터 CL z-map을 구축하는 몇 가지 방법에 대한 비교연구를 다루고자 하며, z-map 자체의 표현 오차에 대해서는 고려하지 않기로 한다. 2절에서는 CL 곡면 정의 및 특이성 문제를 살펴보고, 3절에서 형상 z-map 자체를 기준 모델(master-model)로 이용하는 기존의 CL 곡면 생성방식(3.1절), 매개변수형 곡면모델을 이용하여 기존의 방식을 개선할 수 있는 보완된 방식(3.2 및 3.3절)에 관하여 비교·분석하고자 한다.

#### 2. CL 곡면의 정의 및 특이성 문제

##### 2.1 공구기준점과 CL 곡면

일반적으로 금형 형상부를 표현하는 매개변수형 곡면 모델

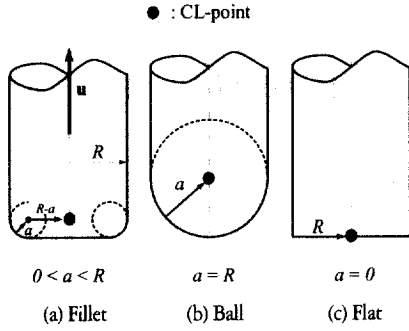


그림 1. 일반적인 공구형상에 대한 CL-point 정의.

은 식 (1)과 같은 벡터함수로 표현되며, 매개변수 영역  $(u, v)$ 에서 3차원 공간  $(x, y, z)$ 로의 정칙사상(regular-mapping)이다.

$$r(u, v) = [x(u, v), y(u, v), z(u, v)] \quad (1)$$

본 논문에서, 곡면  $r(u, v)$ 상의 임의의 공구접촉점에 대한 공구기준점(CL-point)은 <그림 1>과 같이 정의한다(일반적으로 사용하는 공구기준점은  $-au$  만큼 이동한 위치가 된다).

<그림 1>의 공구에 대한 공구기준점의 수학적 표현은 식 (2)와 같다(Choi and Jerard, 1999).

$$r^o(u, v) = r(u, v) + aN + (R - a)(N - \lambda u) / (1 - \lambda^2)^{1/2} \quad (2)$$

여기서,  $\lambda = N \cdot u$ ,

$N$ : unit normal vector at  $(u, v)$ ,

$u = (0, 0, 1)$ : cutter axis,

$R$ : cutter radius (half the diameter),

$a$ : nose radius (ball if  $a = R$ , flat if  $a = 0$ ).

여기서, CL 곡면은 곡면  $r(u, v)$ 에 공구를 접촉한 상태로 공구를 이동하였을 때 공구기준점의 궤적(Choi and Jerard, 1999), 또는 곡면  $r(u, v)$  상에 <그림 1>에서 정의한 공구기준점을 놓고 공구를 이동하였을 때 얻어지는 공구포락면(tool envelope surface, Park et al., 1999)이라 할 수 있다(<그림 2> 참조).

일단 정확한 CL 곡면이 얻어지면 공구경로 생성시 필요한 공구기준점은 CL 곡면 위에 놓이게 된다. 이와 같은 CL 곡면의 수학적 표현이 간단하지 않기 때문에 본 논문에서는 이를 z-map으로 표현하고자 하였다.

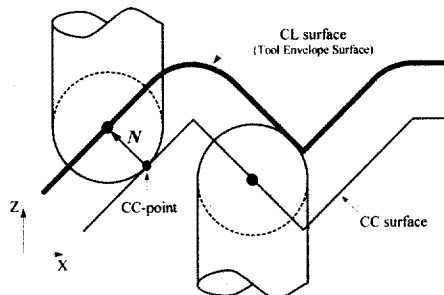
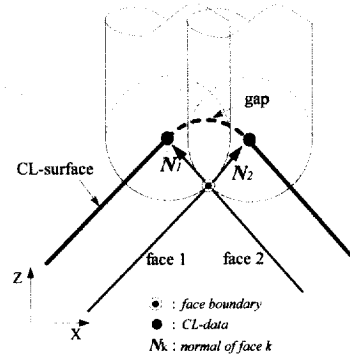
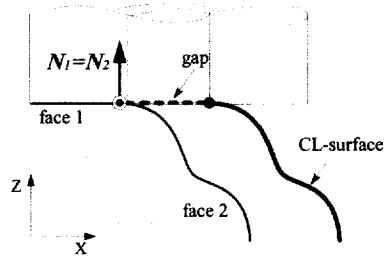


그림 2. 볼엔드밀 공구에 대한 CL-surface.



(a) Ball-end milling case



(b) Flat-end milling case

그림 3. CL 곡면 생성에 있어서의 특이성(singularities) 문제.

### 2.2 특이성 문제

식 (2)는 CL 곡면의 수학적 표현이라 할 수 있으며,  $r^o(u, v)$  또한 벡터함수로서  $(u, v) \rightarrow (x, y, z)$ 의 정칙사상이다. 그러나 공구간섭이 없는 CL 곡면을 구축함에 있어서는 특이성(singularity) 문제를 고려해야 하는 경우가 발생한다. 본 논문에서 ‘특이성’이란  $r(u, v)$  상의 단일 공구접촉점에 대해 단일 공구기준점이 형성되지 않는 경우를 의미하며, 이는 식 (2)로 정의되는 CL 곡면상에서 ‘틈(gap)’으로 나타난다(<그림 3> 참조).

따라서, 공구간섭이 없는 CL 데이터를 생성하기 위해서는 CL 곡면상에 필연적으로 나타날 수 있는 틈에 대한 별도의 처리가 있어야 한다. 다면체 모델의 경우에는 별도의 단위 다면체(예: 삼각형)들로 이러한 틈을 처리해야겠지만, z-map의 경우에는 계산방식 및 데이터 구조 특징상 이런 고려를 할 필요가 없거나 간단한 처리가 가능하다.

역으로 곡면  $r(u, v)$ 의 곡률반경 < 공구반경일 경우 발생하는 ‘꼬임’ 현상에 대해서도 적절한 조치가 필요한데, z-map의 경우 동일한  $(x, y)$ 에 대해 항상 최대 높이값( $z$ )을 취하도록 함으로써 별도의 조치는 필요없다.

### 3. CL z-map 구축

본 논문에서 다루고자 하는 CL z-map 생성 방식은 3가지 기준

모델(master-model)을 사용한다. 기준 모델은 매개변수형 자유 곡면  $r(u, v)$ 로부터 생성한 (i) CC z-map, (ii) CC-곡선, (iii) Offset-facet 모델이다. 첫 번째 기준 모델은 기존의 z-map 역오프셋(inverse offset) 방식에 적용하고, 두 번째 기준 모델은 역모의 가공(inverse cutting simulation) 방식에 적용한다. 마지막 기준 모델은 오프셋 다면체 방식에서 사용하고 있다. 두 번째와 세 번째 기준모델을 활용할 경우, 각 곡면의 경계곡선에 대해서는 공히 역모의 가공을 수행함으로써 2절에서 언급한 특이성 문제를 해결하도록 하였다.

3.1 Z-map 역오프셋 방식

이 방식은 기존의 문헌에 나타난 방식으로써, 형상 z-map(CC z-map)을 사용하여 CL z-map을 구축하는 방식이다. 이 방식은 <그림 4>에 보인것과 같이, CC z-map의 모든 격자점에 공구 기준점을 놓아 가면서 공구의 역형상에 의해 형성되는 공구포락면으로 주위 격자점 높이값을 갱신하는 소위 역오프셋 방식(inverse offset method) 이다(Takeuchi *et al.*, 1989; Choi, 1991; 송수용 & 김석일, 1996; 박정환, 1998a). 이때 CC z-map은 각 격자점(grid-point)에서 정확한 곡면 높이 값을 가지는 기준모델로 생각할 수 있다.

주어진 정밀도( $\delta$ )를 만족하는 CC z-map을 구축하기 위하여 (곡면의 국소적 특징을 평면으로 보고) 식 (3)으로부터 z-map 격자 간격( $\gamma$ )을 계산할 수 있다(<그림 5> 참조). 여기서  $\theta$ 는 곡면의 경사도를 의미하는데, 일반적인 금형 곡면에 대해 이 식을 적용하기 위해서는 최대 경사도를  $\theta$ 로 정하는 것이 의미

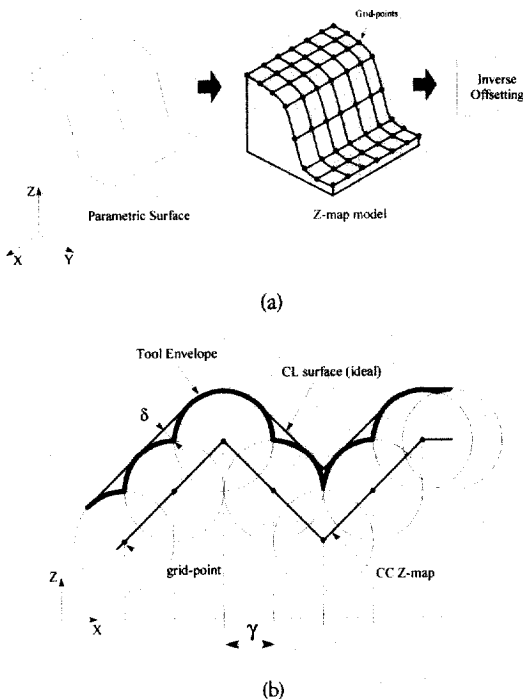
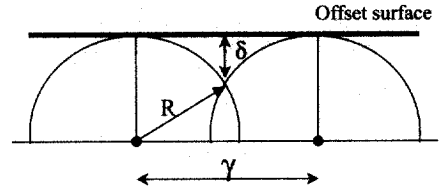
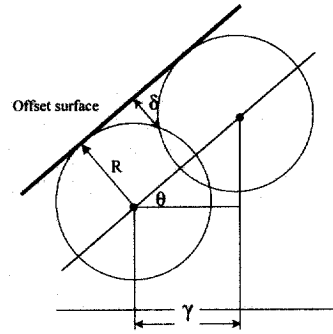


그림 4. 볼엔드밀 공구에 대한 inverse-offsetting.



(a) Horizontal part surface ( $\theta = 0$ )



(b) Slanted part surface ( $\theta > 0$ )

그림 5. CC z-map 격자간격(grid-interval) 결정.

가 있을 것이다.

$$\gamma = 2\sqrt{2R\delta - \delta^2} \cos \theta \tag{3}$$

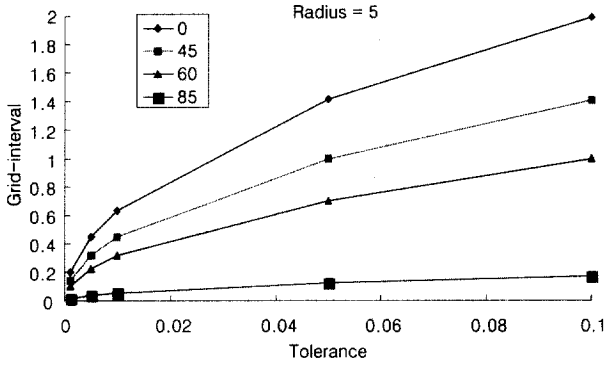
where  $R$  = cutter radius,  $\delta$  = accuracy

<그림 6>의 (a), (b)는 곡면의 경사도 ( $\theta = 0, 45, 60, 85^\circ$ )와 정밀도에 따른 격자간격 크기를 나타낸 예를 보여주고 있다(공구반경  $R = 5, 10$ ). <그림 6>의 (c)에서 보듯이 격자간격( $\gamma$ )은 허용오차(정밀도)와  $\gamma \propto \delta^{1/2}$  관계를 가짐을 알 수 있다.

이 방식은 계산시간 및 계산 강건성(robustness) 측면의 장점이외에도, 특이성 문제를 별도로 고려할 필요가 없다는 장점을 가진다. 그러나 정삭 또는 잔삭 등 정밀가공에 적용할 경우 격자간격을 매우 촘촘하게 주어야 하므로 계산시간이 오래 걸리거나 메모리가 많이 소요되는 단점을 가지고 있다. 참고로, XY 영역 크기가  $1000 \times 1000$ 인 z-map의 단위 격자간격을 1( $\gamma = 1$ )로 구축할 때 소요되는 메모리는 대략 4MB(4byte 실수의 2차원 배열)이다. <그림 6>의 (b)에서 보듯이, 만약 수직벽에 가까운 경사도를 가지는 곡면의 경우( $\theta = 85^\circ$ )라면,  $\delta = 0.005$ ,  $R = 10$ 의 볼 오프셋 곡면을 만들고자 할 때,  $\gamma = 0.055$ 이므로 CC z-map 메모리 소요량은 약 1.3GB임을 알 수 있다. 또한 각 격자점에서 공구 역형상을 놓고 주위 격자점 높이 값을 추출하게 되므로, 격자간격( $\gamma$ )에 대한 공구 직경( $D$ )의 비( $D/\gamma$ )가 작을 경우에는 이상적인 공구포락면과의 오차( $\delta$ : <그림 4>의 (b) 참조)가 커지는 단점을 가진다.

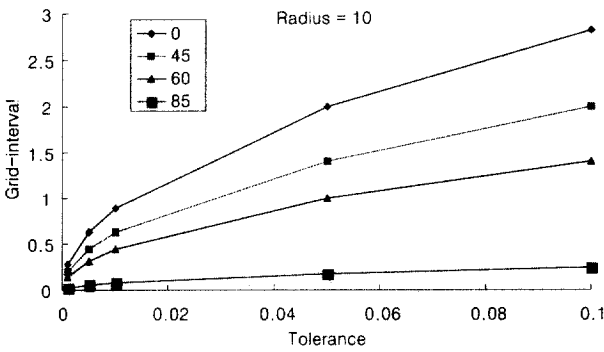
3.2 CC-곡선 역모의가공 방식

3.1절의 방식이 z-map 자체를 '기준모델'로 삼는 데 반해, 본



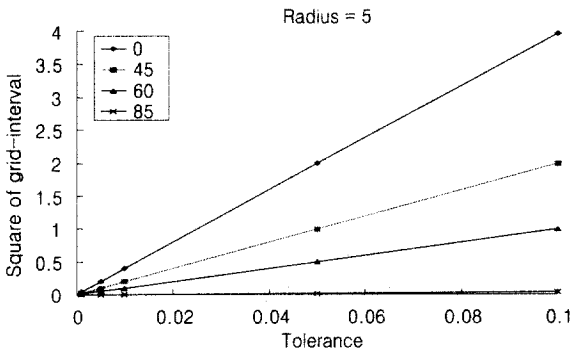
$\delta \backslash \theta$	0	45	60	85
0.001	0.20	0.14	0.10	0.017
0.005	0.45	0.32	0.22	0.039
0.01	0.63	0.45	0.32	0.055
0.05	1.41	0.99	0.71	0.123
0.1	1.99	1.41	0.99	0.173

(a) Cutter radius = 5



$\delta \backslash \theta$	0	45	60	85
0.001	0.28	0.20	0.14	0.025
0.005	0.63	0.45	0.32	0.055
0.01	0.89	0.63	0.45	0.078
0.05	1.99	1.41	0.99	0.174
0.1	2.82	1.99	1.41	0.246

(b) Cutter radius = 10



(c)  $\gamma^2$  vs accuracy

그림 6. 경사각(slant-angle)에 대한 격자간격.

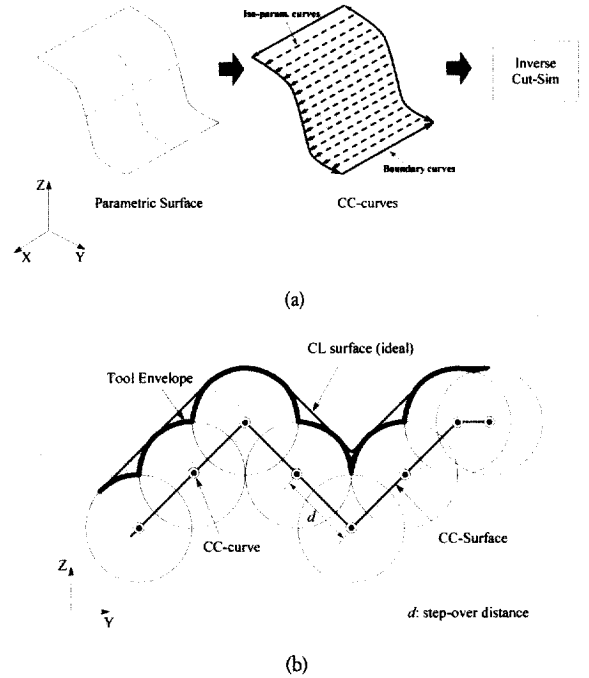


그림 7. 블엔드밀 공구에 대한 inverse cutting-simulation.

방식은 매개변수형 곡면에서 추출한 CC-곡선을 기준모델로 활용한다. 즉 매개변수형 곡면에서 추출한 등경곡선(iso-parametric curve) 및 곡면 경계곡선(boundary-edge curve)을 공구기준점으로 삼아 역모의가공(inverse cutting-simulation) 방식(Choi, 1991; 정연찬, 1996)을 적용하여 CL z-map 모델을 생성한다. 이 방식을 <그림 7>에 나타내었다.

여기서 등경곡선은 점열곡선으로 변환하여 역모의가공을 적용하도록 하며, 이때 점 간격(step-length)은 주어진 가공정밀도( $\delta$ : machining tolerance)를 고려하여 계산한다. 내부 등경곡선 간의 경로간격(step-over distance)에 대해서도 동일한 계산을 적용하여 가공정밀도를 만족하도록 한다. 또한 모든 곡면의 경계곡선으로 역모의가공을 수행함으로써 2절의 특이성 문제를 쉽게 해결할 수 있다. 본 방식의 개략적인 과정은 다음과 같이 정리할 수 있다.

**Procedure CL\_Zmap\_ICS ( $S, T, \epsilon \Rightarrow CLZ[i, j]$ );**

1. Input: trimmed parametric surface  $S$ , z-map  $CLZ[i, j]$ , tool  $T$ , machining tolerance  $\delta$  ;
2. Extract surface edge curves:  $\delta, T, S \rightarrow$  point sequence curves  $EC_i$  ;
3. Extract iso-parametric curves:  $\delta, T, S \rightarrow$  point sequence curves  $IC_j$  ;
4. Perform inverse\_cutting\_simulation & update  $CLZ[i, j] \leftarrow EC_i, IC_j$  ;
5. Return  $CLZ[i, j]$  ;

앞서의 방식이 CC z-map의 격자점 정보만을 활용하는 데 반

해, 본 방식은 곡면 경계곡선 및 내부의 등경곡선을 활용함으로써 z-map '역읍셋 방식'에 비해 더 정확한 기준 모델을 제공한다(예: 꺾인 모서리 형상은 CC z-map 자체에서 유실될 가능성이 크다). 그러나 본 방식은 일반적으로 z-map 역읍셋 방식에 비해 더 많은 계산시간이 소요된다. 이는 z-map 기준 역읍셋 방식의 경우 모든 격자점이  $xy$ -평면상에서 규칙적인 간격을 가지므로 별도의 최적화 기법으로 각 격자점에 대한 높이값 계산영역을 최소화할 수 있는 반면(박정환, 1998a), 본 방식은 등경곡선으로의 역모의가공을 수행함으로써 계산량이 증가할 뿐더러 계산영역의 최적화 효과가 상대적으로 적기 때문인 것으로 사료된다.

또한, 등경곡선 간 간격으로 인하여 CL 곡면상에 커슁(cusp) 형상이 존재할 수 있으며, 이를 보완하기 위하여  $uv$ -영역에서 직교하는 등경곡선에 대해 추가적인 역모의가공을 수행할 경우 계산시간의 증가가 따를 수밖에 없다.

### 3.3 읍셋 다면체 방식

본 방식은 매개변수형 곡면을 삼각형 다면체(facet) 모델로 변환하고, 이로부터 공구에 대한 읍셋 다면체 모델을 만든 후 이 모델로부터 CL z-map의 각 격자점 높이 값을 계산한다. 아울러 각 곡면의 경계곡선에 대해서는 3.2절의 역모의가공을 적용함으로써 특이성 문제를 해결하도록 한다. 본 방식의 개략적인 과정은 다음과 같으며, <그림 8>에 도식적으로 나타내었다.

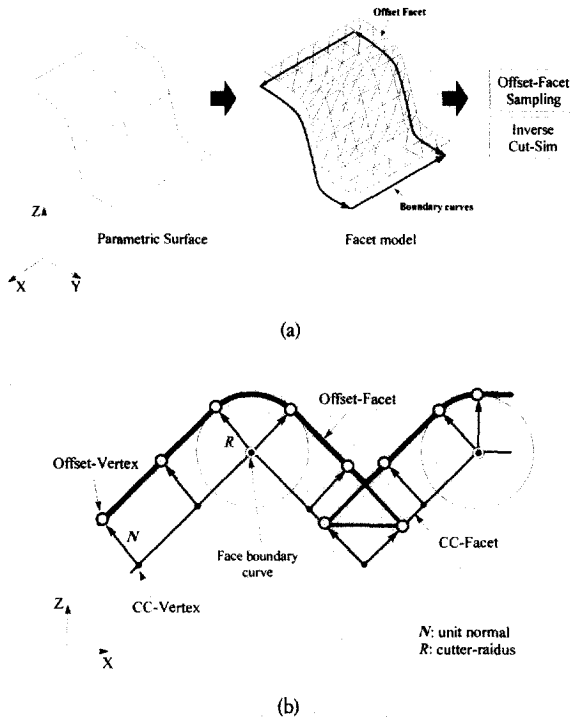


그림 8. 볼랜드밀 공구에 대한 offset-facet sampling & inverse cut-sim.

Procedure CL\_Zmap\_OFS ( $S, T, \epsilon \Rightarrow CLZ[i, j]$ );

1. Input: trimmed parametric surface  $S$ , z-map  $CLZ[i, j]$ , tool  $T$ , tolerance  $\delta$ ;
2. Perform offset triangulation  $\leftarrow S, T, \delta$ ;
3. Update  $CLZ[i, j]$  by individual 'offset triangles' in the triangulated facet-model;
4. Extract surface edge curves:  $\delta, T, S \rightarrow$  point sequence curves  $EC_i$ ;
5. Perform inverse\_cutting\_simulation & update  $CLZ[i, j] \leftarrow EC_i$ ;
6. Return  $CLZ[i, j]$ ;

이때 Step 2에서 생성하는 읍셋 다면체 모델(삼각형)은 단위 삼각형의 각 꼭지점에 저장된 곡면 법선벡터 정보 및 공구정보로부터 식 (2)를 이용하여 계산한다(Choi and Jerard, 1999). 앞서의 두 가지 방식은 정밀도 및 계산시간 측면에서 각각 단점을 가지는 반면, 본 절의 읍셋 다면체 방식은 z-map 역읍셋 방식(3.1절)의 정밀도, 메모리 문제를 보완하면서 CC-곡선 역모의가공 방식(3.2절)에 비해 계산시간을 단축하는 장점을 가진다고 사료된다.

## 4. 적용 사례

3가지 모델에 대해 읍셋반경( $R$ ), 정밀도( $\delta$ )를 달리 주면서 세 가지 방식의 테스트를 수행하였다. 계산시간은 초(sec) 단위이며, 테스트는 Pentium III 500 PC에서 Z-Master NT 버전(Cubictek, 2000)으로 실시하였다.

### 4.1 반구 모델 (convex hemisphere)

<그림 9>에 보인,  $XY$  영역 =  $200 \times 200$ , 곡면개수 = 1, 반경 = 100의 반구에 대해 공구반경( $R$ ) 5, 10, 15의 볼 읍셋을 적용하였고, 정밀도( $\delta$ )를 0.001, 0.005, 0.01로 변화시키면서 계산시간을 측정된 결과를 <그림 10>, <그림 11> 및 <그림 12>에 나타내었다. 여기서 z-map 격자간격은 각각의 정밀도에 대해 0.2, 0.4, 0.6으로 주었으며, 이는 근사적으로  $R$  10,  $\theta$   $45^\circ$ 에 해당하는 격자간격이다. <그림 10>과 <그림 11>에서 Z-map(1)은 z-map 역읍셋을 적용하기 위한 기준모델 (Master-model) z-map을 만드는 시간을 포함하지 아니한 경우이고, Z-map(2)는 포함하는 경우이다. <그림 12>의 (a)는 Z-map(1)의 경우이다.

우선 동일한  $\delta$ 에서  $R$ 의 변화에 따른 계산시간을 살펴보면 (<그림 10> 참조), 읍셋 다면체 방식이 나머지 두 방식에 비해 덜 민감(sensitive)한 것으로 사료된다. 또한  $\delta = 0.001$ 인 경우를 제외하고 거의 모든 경우에 z-map 역읍셋 방식이 가장 빠르며, CC-곡선 역모의가공 방식이 가장 많은 시간을 소요하였음을 알 수 있다. CC-곡선 역모의가공 방식의 경우 다른 두 방식과 달리  $R$  값이 증가함에 따라 계산시간은 오히려 감소함을 알

(b) CC Z-map  
 그림 9. 반구 모델.

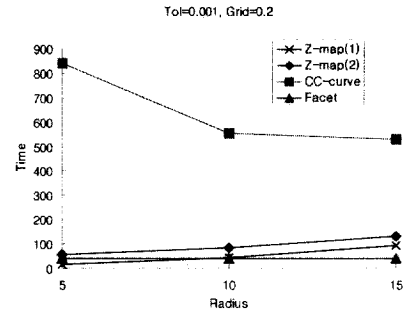
수 있는데(<그림 10>, <그림 12> 참조), 이는 동일한  $\delta$ 에 대해  $R$  값의 증가는 결국 등경곡선의 점간격(step-length) 및 경로 간격(step-over distance)이 증가함으로써 역모의가공(inverse\_cutting\_simulation) 계산량이 줄어들기 때문으로 사료된다.

다음으로 동일한  $R$ 에서  $\delta$ 의 변화에 따른 계산시간을 살펴 보면(<그림 11> 참조), 세 가지 경우 모두 유사하게 반비례 유형을 나타내고 있다. 근사적으로 보아  $\delta$  변화에 대해 CC-곡선 역모의가공 방식이 가장 민감하게 반응하고 있다. <그림 12>의 (a), (b), (c)는 각 방식에 대해  $R$ 과  $\delta$ 를 변화시킨 결과인데, 읍셋 다면체 방식이 3가지  $\delta$ 에서  $R$ 값의 변화에 가장 둔감함을 알 수 있다.

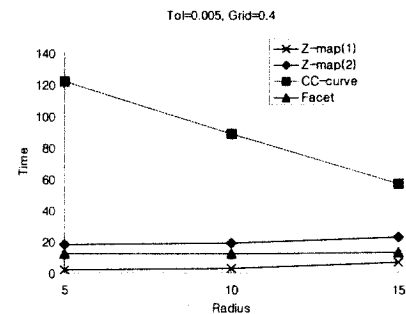
마지막으로 CC-곡선 역모의가공과 읍셋 다면체 방식의 경우 CC z-map이 아니라 CL z-map의 격자간격이 계산시간에 영향을 미치므로, 다른 조건을 동일하게 두고( $\delta = 0.005, R = 10$ ) CL z-map의 격자간격을 0.2~1.0으로 변화시켰을 때 계산시간을 관찰하였다(<그림 12>의 (d) 참조). 이 경우에도 읍셋 다면체 방식이 다른 방식에 비해 비교적 둔감한 변화를 보임을 알 수 있다.

4.2 Crank 모델

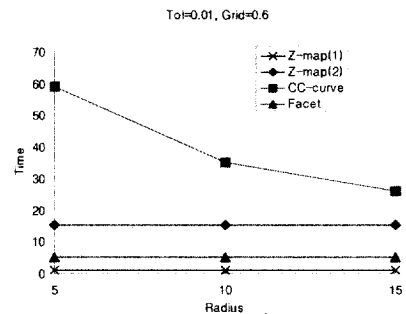
<그림 13>의 XY 영역= $105 \times 110$ , 곡면개수=6인 모델에 대해 z-map 격자간격( $\gamma$ )을 0.35로 하여, 공구반경( $R$ ) 1~25의 불 읍셋을 적용하였다. 정밀도( $\delta$ )를 0.001, 0.005, 0.01, 0.05, 0.1로 변화시키면서 계산시간을 측정된 결과를 <그림 14>에 나



(a)  $\delta = 0.001$

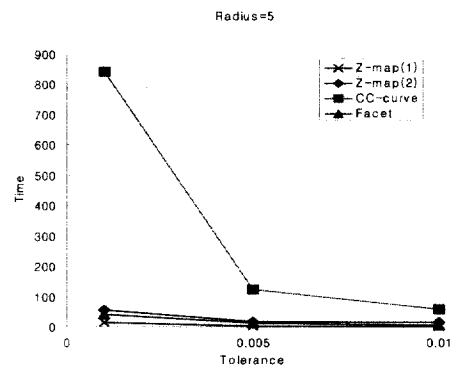


(b)  $\delta = 0.005$

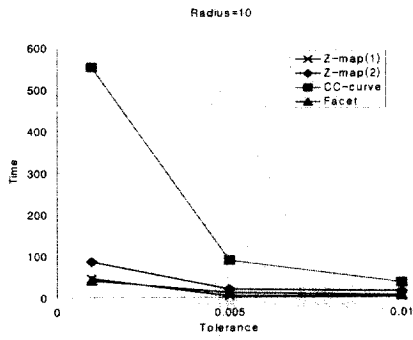


(c)  $\delta = 0.01$

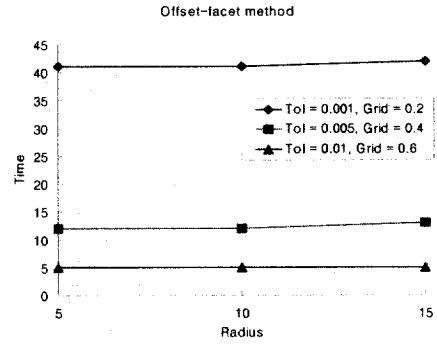
그림 10. 반구 모델: 읍셋 반경( $R$ ) 변화에 대한 계산시간 비교 ( $\delta$  고정).



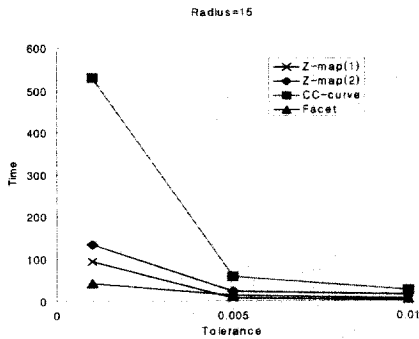
(a)  $R = 5$



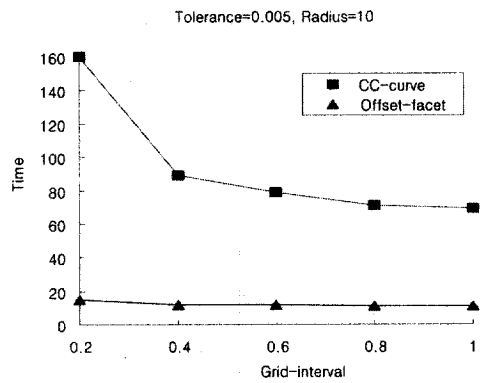
(b) R = 10



(c) Offset-facet sampling



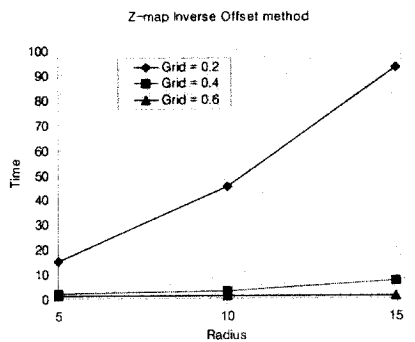
(c) R = 15



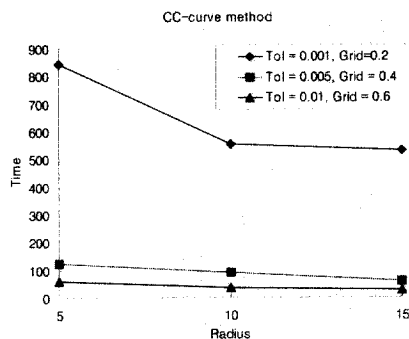
(d) For various CL Z-map grid-interval

그림 11. 반구 모델: 정밀도( $\delta$ ) 변화에 대한 계산시간 비교(R 고정).

그림 12. 반구 모델: 3가지 방식의 계산시간 비교.



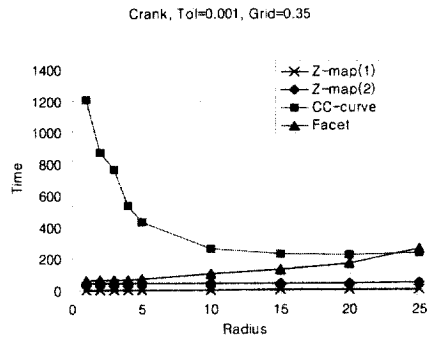
(a) Z-map inverse offset



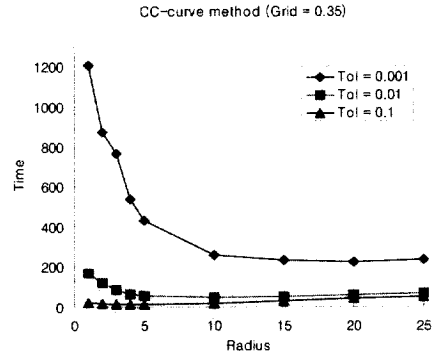
(b) CC-curve inverse cutting-simulation

(b) CC Z-map

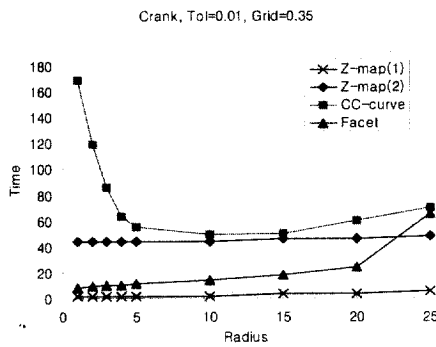
그림 13. Crank 모델( $\gamma = 0.35$ ).



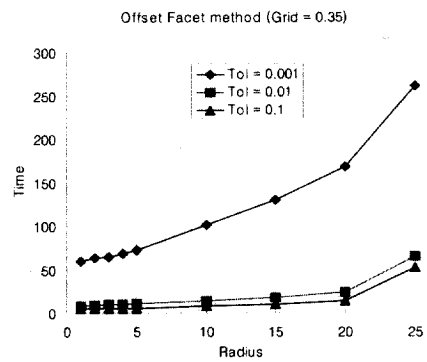
(a)  $\delta = 0.001$ , grid-interval = 0.35



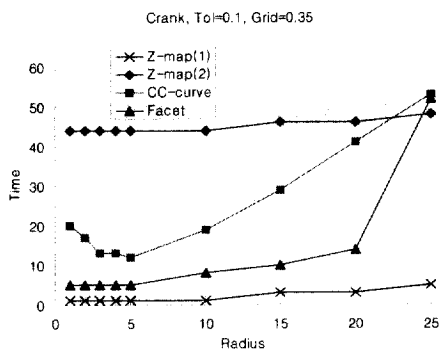
(a) CC-curve inverse cutting-simulation



(b)  $\delta = 0.01$ , grid-interval = 0.35



(b) Offset-facet sampling



(c)  $\delta = 0.1$ , grid-interval = 0.35

그림 14. Crank 모델: 읍셋 반경( $R$ ) 변화에 대한 계산시간 비교 ( $\delta$  고정).

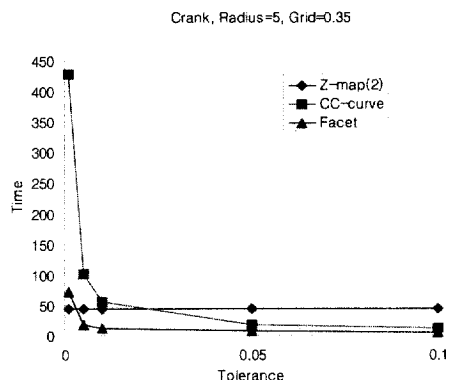
타내었다. 여기서  $\gamma$ 는 변화하지 않았기 때문에 z-map 역읍셋 계산시간은 모두 동일하다.

앞서의 반구형상과는 달리 CC-곡선 역모의가공 방식에서  $\delta = 0.01, 0.1$ 의 경우(<그림 14>의 (b), (c) 및 <그림 15>의 (a) 참조),  $R$ 이 증가함에 따라 계산시간이 오히려 증가하는 양상이 관찰되었다. 이는 동일한  $\delta$ 에서  $R$ 이 증가하면 역모의가공할 CC-곡선의 개수와 점의 개수는 줄어들지만,  $R$ 의 증가로 인해 z-map 수정영역(update-region)이 증가하면서 계산시간이 오히려 늘어난 것으로 사료된다. 반면, <그림 14> 및 <그림

그림 15. Crank 모델: 두 가지 방식의 읍셋 계산시간 비교.

15>의 (b)를 살펴보면, 읍셋 다면체 방식은  $R$ 값의 범위 1~20 사이에서 계산시간이 거의 선형적으로 증가함을 관찰할 수 있다.

<그림 16>은  $\delta$ 의 변화(0.001~0.1)에 대한 계산시간의 추이를 보여준다.  $R$ 값 5, 10, 15에 대해 거의 유사한 패턴임을 알 수 있으며, 앞서의 반구모델과도 다르지 않다고 보여진다. 참고로 여기서의 z-map 역읍셋 계산시간은 CC z-map을 계산하는 시



(a)  $R = 5$



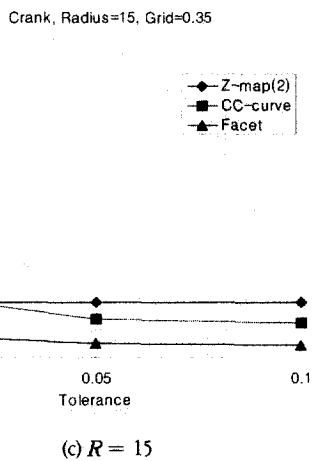
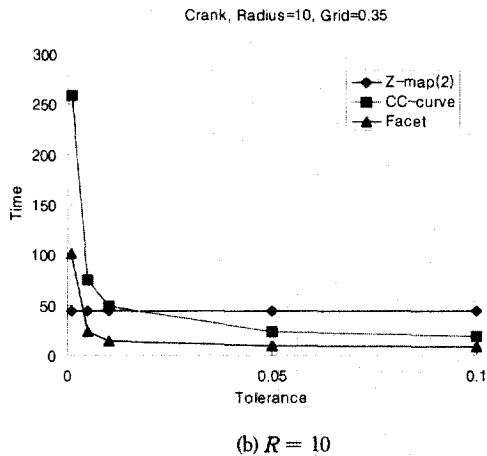


그림 16. Crank 모델: 정밀도( $\delta$ ) 변화에 대한 계산시간 비교( $R$  고정).

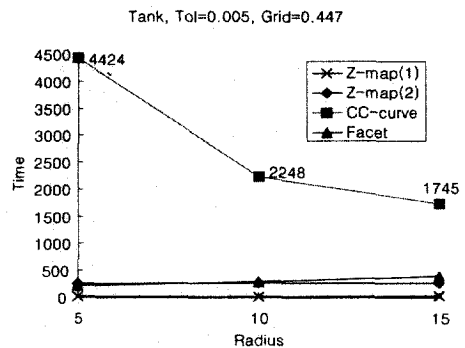
간을 포함한 경우이다.

4.3 연료탱크 모델(car fuel tank model)

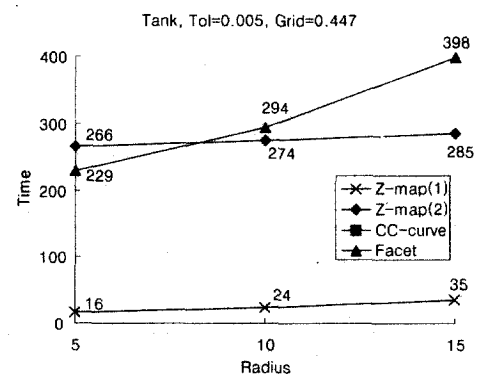
<그림 17>에 보인,  $XY$  영역 =  $1190 \times 890$ , 꼭면개수 = 1880 인 자동차 연료탱크 모델에 대해 읍셋반경( $R$ ) 5, 10, 15로 세 가지 방식을 테스트하였다( $\delta = 0.005$ ). 여기서 z-map 격자간격( $\gamma$ )은 0.447로 하였는데, 이는  $\delta = 0.005$ ,  $R = 10$ ,  $\theta = 45^\circ$ 에 해당하는 격자간격이며(<그림 6> 참조), CC Z-map의 메모리 소요량은 약 21.2MB이다. 참고로  $\theta = 85^\circ$ 에 해당하는  $\gamma$ 의 경우라면 약 1.5GB의 z-map 메모리가 필요할 것이다.

<그림 18>의 (a)는 세 가지 방식의 계산시간을 나타내며, CC-꼭선 역모의가공을 제외한 두 가지 방식의 계산시간을 <그림 18>의 (b)에 보이고 있다. 그림에서 알 수 있듯이 master-model z-map을 생성하는 데 필요한 계산시간을 제외한다면, z-map 역읍셋 방식이 읍셋 다면체 방식보다 계산시간이 적게 소요된다. 읍셋반경 증가에 따른 계산시간의 변화는 앞의 두 예제와 크게 다르지 않은 것으로 관찰된다.

(b) CC z-map  
그림 17. 연료탱크 모델( $\gamma = 0.447$ ).



(a) Computation time of the three methods



(b) Magnified view of chart (a)

그림 18. 연료탱크모델: 읍셋 계산시간 비교.

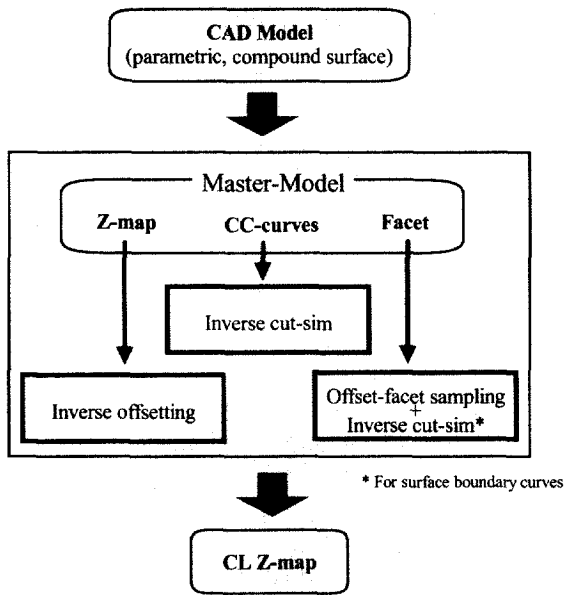


그림 19. 오프셋 방식의 요약.

### 5. 요약 및 분석

#### 5.1 3가지 방식의 요약

3절에서 설명한 세 가지 CL Z-map 구축 방식을 정리하면 <그림 19>와 같다. 매개변수형 복합곡면(parametric compound surface)으로 모델링한 형상곡면이 주어졌을 때, CL 곡면을 생성하기 위한 기준모델을 CC z-map(형상곡면 z-map), CC-곡선(형상곡면의 등경곡선 및 경계곡선), 그리고 삼각형 다면체(offset facet 및 곡면 경계곡선) 모델로 구축하였다. 또한 CL 곡면은 특이성 문제해결이 간단한 z-map 모델로 표현하도록 하였다.

각 기준모델로부터 CL z-map을 구축하는 데 필요한 프로그램 기능(function)은 각각 inverse offsetting 기능, inverse cutting-simulation 기능, 그리고 offset-facet sampling 및 inverse cutting-simulation 기능이다. 본 연구에서는 Z-Master CAM 시스템의 내부 기능을 일부 활용하였다.

#### 5.2 테스트 결과의 분석

4절에서 수행한 3가지 테스트 결과를 기초로 각 방식의 특성을 정성적으로 <표 1>에 정리하였다. <표 1>에서는 가공 허용오차( $\tau$ : machining tolerance)를 정밀도( $\delta$ )와 동일한 의미로 사용하였다.

CL z-map 계산시간은 일반적으로 z-map 역오프셋 방식이 가장 우수한 반면, 기준모델을 구축하는 데 필요한 메모리 소요량은 이 방식이 가장 불리하다. 만약 주어진 형상곡면으로부터 CC z-map, 즉 master-model z-map을 구축하는 시간을 포함한다면, 오히려 오프셋 다면체 방식이 더 빠른 경우도 있음을 확인할

표 1. 3가지 오프셋 방식의 정성적 비교

	CC z-map inverse offset	CC-curve inverse cut-sim	Offset-facet sampling
Computation time (Factors)	○ ( $\gamma_C \equiv \tau, R, \gamma_L$ )	× ( $\tau, R, \gamma_L$ )	△ ( $\tau, R, \gamma_L$ )
Memory requirement	×	○	○
Accuracy of reference model with special features*	×	△	○

$\gamma_C$  = CC z-map grid-interval,  $\gamma_L$  = CL z-map grid-interval,  
 $\tau$  = given machining tolerance (accuracy),  
 $R$  = offset distance (cutter radius)  
 \*sharp edges, vertical wall

수 있었다. 본 연구에서 적용한 세 가지 모델에 대한 테스트 결과를 기초로, 각 방식의 계산시간( $t_1, t_2, t_3$ )은 식 (4)~(6)와 같이 표현할 수 있을 것이다.

1. Z-map inverse offset:  $t_1 = f(1/\gamma_C, R, \gamma_L)$  or  $t_1 = f(1/\tau^{1/2}, R, \gamma_L)$  (4)

2. CC-curve inverse cut-sim(1):  $t_2 = g_a(1/\tau, 1/R, \gamma_L)$  (5-a)

3. CC-curve inverse cut-sim(2):  $t_2 = g_a(1/\tau, 1/R, \gamma_L)$  (5-b)

4. Offset-facet sampling:  $t_3 = h(1/\tau, R, \gamma_L)$  (6)

식 (4)는  $\gamma_C \propto \tau^{1/2}$  (<그림 6>의 (c) 참조)로부터 알 수 있으며, 식 (5)의 경우는 <그림 14>의 (b), (c)로부터 유추하였다. 일반적으로 z-map 역오프셋 방식에서 계산시간은  $(R/\gamma)^2$ 에 비례한다(Choi, 1991). 그러나 본 연구에서는 각 격자점에서의 계산영역을 최적화 함으로써 대략  $R/\gamma$ 에 비례하는 결과를 관찰할 수 있었다. CC-곡선 역모의가공 방식의 경우, 동일한  $\delta$ 에서 계산시간은  $R$ 에 반비례하나(식 5-a: <그림 10>, <그림 12>의 (b), <그림 14>의 (a), <그림 18> 참조), 경우에 따라서는  $R$ 에 비례하는 경우가 관찰된다(식 5-b: <그림 14>의 (b), (c) 참조). 이론적으로 동일한  $\delta$ 에서  $R$ 이 증가하면 역모의가공할 CC-곡선의 개수와 점의 개수는 줄어든다. 그러나  $R$ 의 증가로 인해 z-map 수정영역(update region)이 증가하면서 계산시간이 오히려 늘어난 것으로 사료된다

4절의 3가지 예에 대한 테스트 결과를 종합적으로 따져보았을 때, 동일한 가공허용오차( $\tau$ ) 및 공구반경( $R$ )에서 CL z-map 계산시간은 일반적으로 CC-곡선 역모의가공 방식이 가장 불리하며, 다음은 오프셋 다면체 방식과 z-map 역오프셋 방식의 순서가 될 것으로 사료된다. 참고로, CC-곡선 및 오프셋 다면체를 기준모델로 사용할 경우, CL z-map 격자간격이 증가함에 따른 계산시간 변화량은 오프셋 다면체 방식이 훨씬 안정적임을 알 수 있다(<그림 12>의 (d) 참조).

<표 1>에서 언급한 기준모델의 정확도(accuracy of reference model)에 대해서는 추가적인 설명이 필요하다. 즉 형상면을 z-map으로 표현하는 경우(CC z-map), 가장 큰 문제는 수직벽 및

꺾인 모서리(곡면간 연결부위가 G'연속을 만족하지 않는 경우) 등의 특징형상 표현이 어렵다는 점이다(박정환 *et al.*, 1998b). 어느 정도는 격자간격을 작게 주어 해결할 수 있겠지만, 메모리 소요량을 따져볼 때 현실적이지 못한 경우가 발생할 수 있다. 또한 이 방식과 두 번째 방식에서도 방법의 특성상 CL 곡면상에 커슴형상을 만들 가능성이 높으며, 이는 주어진 정밀도를 만족하더라도 결코 바람직한 결과는 아니라고 사료된다.

6. 결론 및 토의

본 연구에서는 세 가지 기준모델을 이용하여 z-map 모델로 공구포락면(CL 곡면)을 구축하는 방법을 살펴보았으며, 정밀도( $\delta$ ) 및 읍셋반경( $R$ ), 격자간격( $\gamma$ )을 변화시켰을 때 각 방식의 상대적인 CL z-map 계산시간을 비교, 분석하고자 노력하였다. 특히 z-map 모델은 공구포락면을 구축함에 있어서 고려해야 하는 특이성 문제(2절)를 쉽게 해결할 수 있으며, 읍셋팅이나 높이값 계산 등이 단순하다는 장점을 가지고 있다.

계산시간, 메모리 소요량 및 형상곡면 표현 정확도 등을 고려할 때, 적용분야에 따라 z-map 역읍셋 방식과 읍셋 다면체 방식을 적절하게 활용하는 것이 바람직할 것이다. 즉, 가공 허용오차 및 공구 크기가 상대적으로 크면서 가공여유가 있는 황삭/중삭 가공경로 생성에는 z-map 역읍셋 방식을 사용할 수 있으며, 정삭 또는 잔삭 가공에 대해서는 읍셋 다면체 방식이 적합한 것으로 사료된다.

한편, z-map(CC z-map, CL z-map) 자체의 표현오차를 줄이는 방법으로는 EZ-map 등의 별도 보완 모델(박정환 *et al.*, 1998b)을 사용하는 것이 가능하며, 본 연구에서 언급한 방식을 쉽게 적용할 수 있다. EZ-map 모델의 구조는 <그림 20>에 나타내었는데, 기존의 z-map 격자점 사이에 별도의 간격으로 격자점을

추가함으로써 형상을 더 정밀하게 나타낼 수 있는 모델이다.

참고로, Z-Master 2000 CAM 시스템(Cubictek, 2000)에서는, 황삭이나 중삭 가공 데이터 생성시 z-map 또는 EZ-map의 역읍셋 방식을 사용하며, 정삭 및 잔삭 가공데이터 생성에는 EZ-map 기반의 읍셋 다면체 방식을 채용하고 있다. 그러나 형상곡면의 CAD 모델이 존재할 경우에는, 특징형상을 포함하는 형상곡면 표현 정확도를 고려하여 가능한 읍셋 다면체 방식을 사용하는 것을 원칙으로 하고 있다. 또한 미국 크라이슬러 자동차 회사의 스티로폼 모형(pattern) 제작과정에서는 z-map 역읍셋 방식으로 가공 데이터를 생성하고 있다(박정환, 1998a).

참고문헌

박정환 (1998a), 자동차 프레스금형의 스티로폼-패턴 가공을 위한 전용 CAM 시스템 개발, *한국CAD/CAM 학회 논문집*, 3(4), 223-235.  
 박정환, 정연찬, 최병규 (1998b), Z-map 모델을 이용한 정밀형상 모델링, *한국정밀공학회지*, 15(11), 180-188.  
 송수용, 김석일 (1996), Z-map 모델을 이용한 3차원 CNC 가공계획 및 절삭시뮬레이션에 관한 연구, 13(5), 115-121.  
 정연찬 (1996), 금형가공용 통합 CAM 시스템을 위한 가공데이터의 모의가공과 검증에 관한 연구, 박사학위논문, 한국과학기술원.  
 Choi, B. K. (1991), *Surface Modeling for CAD/CAM*, Elsevier.  
 Choi, B. K., Chung, Y. C., Park, J. W. and Kim, D. H. (1994), Unified CAM-system Architecture for Die and Mould Machining, *Computer-Aided Design*, 26(3), 235-243.  
 Choi, B. K. and Jerard, R. B.(1999), *Sculptured Surface Machining*, Kluwer Academic Publishers.  
 Choi, B. K., Chung, Y. C. and Park, J. W. (1995), Application and Extension of Z-map Model, *Proc. of the Pacific Conference on Computer Graphics and Applications*, Seoul, 221-234.  
 Cubictek (2000), *Z-Master 2000 User's Manual*, Cubictek, Korea.  
 Do Carmo, M. (1976), *Differential geometry of curves and surfaces*, Prentice-Hall.  
 Hwang, J. S. (1992), Interference-free Tool-path Generation in the NC Machining of Parametric Compound Surfaces, *Computer-Aided Design*, 24(12), 667-676.  
 Lee, S. X. and Jerard, R. B. (1994), 5-axis Machining of Sculptured Surfaces with a Flat-end Cutter, *Computer-Aided Design*, 26(3), 165-178.  
 Park, J. W., Kim, B. H. and Choi, B. K. (1999), Pencil curve tracing via virtual digitizing, *Machining Impossible Shapes*, Kluwer Academic Publishers, 279-292.  
 Saito, T. and Takahashi, T. (1991), NC Machining with G-buffer Method, *Computer Graphics*, 25(4), 207-216.  
 Takeuchi, Y., Sakamoto, M., Abe, Y. and Orita, R. (1989), Development of a Personal CAD/CAM System for Mold Manufacture Based on Solid Modeling Techniques, *Annals of the CIRP*, 40, 455-458.

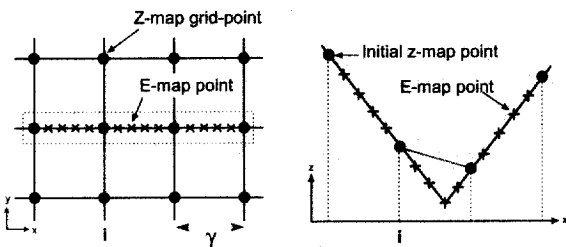


그림 20. EZ-map model의 구조.