大韓機械學會論文集 第18卷 第1號, pp. 53~64, 1994

〈論 文〉

옥트리를 이용한 황삭 가공경로생성

김태주* · 이건우** · 홍성의***

(1993년 5월 10일 접수)

Tool Path Generation for Rough Cutting Using Octree

Tae Joo Kim, Kunwoo Lee and Sung Eui Hong

Key Words: Rough Cutting(황삭), NC Milling(NC 밀링가공), Tool Path Generation(공구경로생성) CAM(캠), Octree(옥트리), Octant(팔분체)

Abstract

Rough cutting process takes the major portion of machining operation using NC milling machine. Especially, most of the machining time is spent in this process when molds are machined. Therefore, an efficient algorithm for generating the tool path for rough cutting is suggested in this paper. The first step of the procedure is getting the volume to be machined by applying the Boolean operation on the finished model and the workpiece which have been modeled by a solid modeling system. Basic principle of determining machining procedure is that a large tool should be used at the portion of the simple shape while a small tool should be used at the complex portion. This principle is realized by representing the volume to be machined by an octree, which is basically a set of hexahedrons, and matching the proper tools with the given octants. When the tools are matched with the octants, the tool path can be derived at the same time.

1. 서 론

최근 들어 NC 공작기계의 보급과 더불어, NC 가공을 위한 공구경로를 자동으로 생성하는 소프트웨어에 관한 연구와 그 개발이 활발히 진행되고 있다. NC 가공은 보통 황삭, 중삭, 정삭과정을 거치는데, 이 중에서 정삭과정은 현재 많은 연구가 수행되었으나[1~6] 황삭 알고리즘(algorithm)에 관한 연구는 상대적으로 미진한 편이다. 그러나, 특히 금형가공에 있어서는 금속괴로부터 완전한 형상을 절삭하는데, 황삭에서 제거해야 할 체적이 전체 금속괴의 약 70%나 되는 것이 많고, 절삭시 소요되는 시간도 크다. 따라서, 효율적인 황삭 알고리즘이 필요하다.[11]

황삭 알고리즘은 다음과 같이 크게 두 가지 형태로 나눌 수 있는데, Fig. 1은 그 두 형태를 보여주고 있다. 첫번째로, Fig. 1(a)의 경우는 피삭재가 이미 주조 등으로 완성품의 형상을 어느 정도 갖추고 있는 경우에 사용하는 방법이다. 이와 같은 경우에는, 황삭 공구경로를 비교적 쉽게 생성할 수 있다. 즉, 정삭 공구경로를 일정한 거리만큼 곡면의 법선방향으로 옵셋(offset)시키면 황삭 공구경로가 된다. 두번째로는 Fig. 1(b)와 같이 피삭재의 초기형상이 완성품과 관계없이 금속괴로 주어질 경우이며, 금형가공시에 많이 나타난다. 이때는, 일정한 깊이만큼 내려가면서 절삭한 후, 남아있는 계단형태 부분은 첫번째 방법과 같은 방법으로 제거하게 된다. 이 경우에는 첫번째 경우보다 황삭할

*대우중공업(주) 기계기술부
**정회원, 서울대학교 기계설계학과
***서울대학교 기계설계학과 대학원

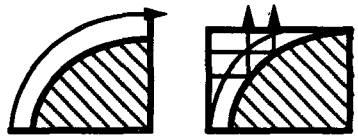


Fig. 1 Two types of rough cutting

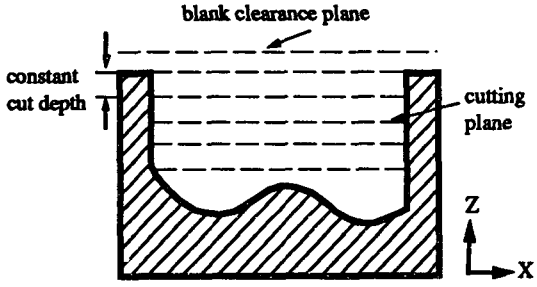


Fig. 2 A method used in a comercial system

양이 많아 황삭에 소요되는 시간도 자연히 많게 되고, 또한 공구경로생성도 전자의 경우보다 더 복잡하다. 따라서 본 연구에서는 Fig. 1(b)와 같은 형태에 대한 효율적인 황삭가공 알고리즘을 개발하고자 한다.

황삭기능을 제공하는 상용 CAD/CAM시스템[7,8]은 Fig. 1(b)와 같은 형태에 대한 기능을 제공하고 있다. 예를 들면, Fig. 2는 Unigraphics II[7]에서 사용한 방법을 보이고 있다. 이들은 공구의 움직임이 물체에 전혀 간섭을 주지 않는 평면(clear plane)을 사용자로부터 입력받아, 이 평면으로부터 일정한 깊이만큼씩 사용자가 제시한 공구를 가지고 차례차례 절삭하도록 하고 있다. 이때 절삭면(cutting plane)은 z=constant인 평면이며 다음 절삭면까지의 깊이가 일정하다.

그런데, 이와 같은 방법을 사용하면, 복잡한 형상의 물체를 절삭할 경우, 간섭을 피하기 위해 공구의 크기가 작아져야 한다. 따라서, 필요 이상으로 작은 공구로 절삭하는 부분이 많이 발생하게 되고, 이로 인한 소요시간도 길다. 또한 작은 공구로 많은 시간을 절삭하면, 공구의 마멸때문에 오차가 커지고, 마멸로 인한 공구 교환문제가 발생하게 된다.

Yamaguchi[9]와 Yuen[10]은 황삭가공에 옥트리(octree)를 적용한 알고리즘을 연구했는데, 이들은 가공해야 할 형상으로부터 주어진 정밀도를 만족하게끔 옥트리를 생성한 후 앞의 상용시스템과 같은 방법으로 사용자에 의해 주어진 공구를 갖고, 일정한 깊이만큼 차례차례 내려가면서 절삭한다. 그런데 이 방법은 가공형상으로 옥트리모델을 이용한다는 것 외에는 앞의 상용시스템과 차이점이 없기 때문에 마찬가지로 공구선정에 따라 많은 절삭시간을 소요할 수 있다는 단점을 가진다.

위와 같은 단점을 개선하기 위하여 이계정[11]은 지름이 큰 공구와 작은 공구를 사용할 수 있게 한 알고리즘을 개발하였다. 이 방법은 먼저, 앞의 상용시스템에서와 같이 주어진 공구로 절삭면을 구한 다음, 절삭시에는 주어진 공구지름의 2배에 해당하는 공구를 갖고 절삭면 2개에 해당하는 부분을 먼저 가공한 뒤 남은 부분을 주어진 공구로 가공함으로써 절삭성 향상을 꾀하였다. 하지만 사용자가 선정한 공구가 절삭 모델의 형상에 접합치 않을 경우에는 오히려 절삭 효율이 저하될 우려가 있다.

한편, Lee Yuan Shin[12]는 곡면의 근사정보로부터 공구를 찾아내지만 찾는 공구가 곡면을 모두 가공할 수 있는 최대 공구 한 가지만을 찾아내므로 단지 상용시스템에서 공구 입력만을 자동화한 것뿐이다.

따라서, 본 연구에서는 위의 연구들의 단점을 극복하기 위하여, 솔리드모델러로 모델링한 물체의 정보로부터, 기하학적인 조건을 고려하여, 사용가능한 공구들을 공구 데이터베이스(tool database)로부터 자동으로 선택해 내고, 큰 공구부터 순차적으로 적용하여, 절삭효율을 높여 주는 공구경로생성 알고리즘을 개발하였다. 이를 위하여, 현재 서울대 CAD연구실에서 개발한 솔리드모델러(solid modeler)인 SNUMOD를 이용하여, 원재료의 형상과 가공이 끝난 완성품의 형상을 모델링하고, 공구의 선택과 공구경로의 생성은 옥트리분할기법을 적용시켜 생성된 팔분체(octant)를 참조하여 이루어지도록 하였다.

2. 옥트리의 개요

옥트리(octree)는 공간을 여러크기의 육면체로 분할하여 물체를 표현하는 기법으로서 물체의 모델링(modeling), image processing, ray tracing, 유한요소생성 등 다양한 용도로 쓰이고 있다.[13~15] 옥트리 생성은 우선 주어진 모델을 모두 포함하는 최소 크기의 육면체(minmax box)를 먼저 만들고,

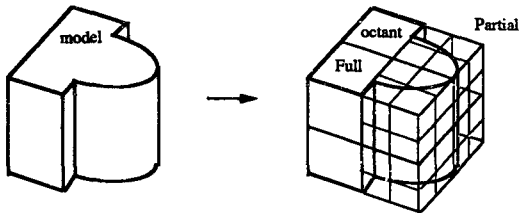
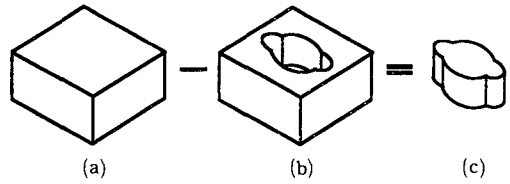


Fig. 3 Generation of an octree



A : Workpiece model
B : Part model
C : Volume to be machined

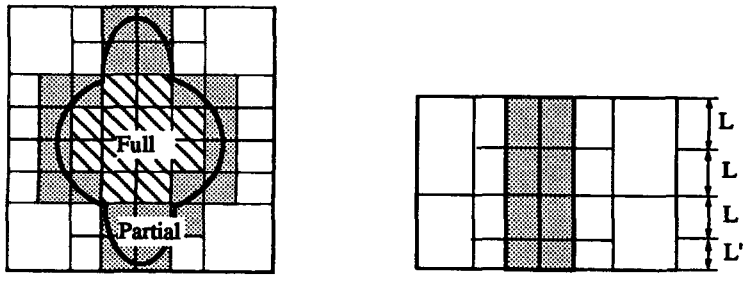
Fig. 4 Generation of volume to be machined

이를 분할하게 된다. 옥트리의 분할은 *xy, xz, yz*평면에 의해 각각 이등분하므로 한번 분할할 때마다 8개의 육면체 요소가 생기게 된다. 이렇게 생긴 육면체 요소 하나 하나를 팔분체(octant)라 하며 생성된 각 팔분체는 모델과의 포함관계에 따라 Full, Partial 및 Empty 형으로 구분된다. 각 방향으로의 분할은 최소 팔분체의 크기가 허용크기가 될 때까지 Partial형 팔분체만을 계속 분할하고, 나머지 팔분체들은 분할하지 않는다. 이러한 분할을 통한 옥트리 생성의 예를 Fig. 3에 보였다.

3. 옥트리를 사용한 황삭 공구경로생성

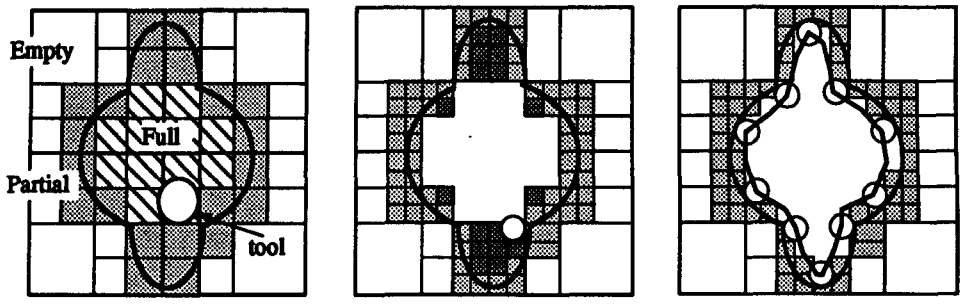
본 연구에서는 황삭에서의 절삭시간을 줄이기 위해 큰 공구를 선택하여 대략 절삭한 다음, 작은 공구로써 남은 국소의 복잡한 부분을 절삭하게 하는 방법을 택하였다. 이때, 어떤 공구로 얼마만큼을 절삭할 것인지를 결정하는 것이 문제점이 되는데, 이것은 앞의 2장에서 기술한 옥트리의 성질을 이용하여 해결하였다. 즉, 먼저 몇번의 옥트리 분할로 Full형 팔분체가 생기면 더 이상 잘게 분할하기 전에 큰 공구를 이용하여 Full형 팔분체를 절삭하게 한 다음, 남은 Partial부분에 대해 분할을 계속하여, 이때 새로 생성된 Full형 팔분체를 작은 공구로 절삭하게 한다. Fig. 4에서 Fig. 6까지의 그림들이 그 과정을 보여주고 있는데 이를 개략적으로 설명하면 다음과 같다.

Fig. 4에서 우선 사용자가 금속괴 모델 A와 최종 제품모델 B를 모델러를 사용해 입력하면 불리언 작업(boolean operation)중 빼기(subtraction)를 적용하여 절삭해야 할 부분인 C의 모델을 얻어낸 후 이에 대해 옥트리를 생성하게 된다. 따라서, 생성된 옥트리내의 Full형 팔분체들이 우선적으로 절삭해야 할 대상이 된다. 불리언작업으로 구한 절삭모델 C에 옥트리를 생성시키면 Fig. 5와 같이 된다. 여기서 특이한 점은 Partial형 팔분체의 분할은 가로, 세로로는 항상 이등분되어 분할되어진 각각의 가로, 세로의 길이는 같으나, 높이는 가로, 세로와 일정한 비를 유지하고 있기 때문에 높이가 가로, 세로보다 일정비 이하이면 분할되어지지 않는다는 것이다. 또한 항상 이 비율을 유지하게끔 분할하기 때문에 Fig. 5에서처럼 가장 아래층을 분할할 때는 분할되어 새로 생성될 8개의 팔분체 중 아래 4개의 높이 L'가 위의 4개의 높이 L보다 작아질 수 있다. 즉, 위의 팔분체의 높이는 가로, 세로와 일정



Front view Top view

Fig. 5 Generation of an octree



(a) Selection of the maximum tool from full type octant
(b) Result after the first cutting
(c) Profiling

Fig. 6 Rough cutting process using octree

비를 계속 유지하고, 아래 팔분체는 분할되기 전의 팔분체 높이에서 위의 팔분체 높이를 뺀 길이가 된다. 이러한 분할은 3.2절에 자세히 언급되어 있다.

다음으로 공구를 선택하여야 하는데 Fig. 6(a)처럼 절삭모델 C에 생성된 옥트리의 팔분체 중 우선 Full형을 절삭하기 이해 가장 작은 Full형 팔분체에 대응되는 공구를 선택하면, 이것이 최대 공구가 된다. 이렇게 선택된 공구로 위층부터 차례로 Full형 팔분체를 절삭하여 아래층까지 모두 절삭한다. 다음에는 절삭되지 않고 남은 Partial형 팔분체를 한 단계 더 분할하여 Fig. 6(b)와 같은 결과를 얻게 된다. 앞에서와 마찬가지로 이번에도 가장 작은 Full형 팔분체에 대응되는 공구를 찾아 이전 공구와 교환한 뒤 모든 층을 내려가면서 작아진 Full형 팔분체를 절삭하게 한다. 이 과정을 반복하다가, 공구가 전 곡면을 모두 가공할 수 있도록 충분히 작아지면 Fig. 6(c)와 같이 경계에 조금 남아 있는 부분을 각 층을 따라가면서 프로파일링(profiling)으로 완전히 제거한다.

이러한 방법으로 큰 팔분체에서 작은 팔분체에 이르는 일련의 Full형 팔분체를, 거기에 맞는 공구를 선택하여 가공함으로써, 복잡하지 않는 부분은 큰 공구를 사용하고, 남은 국소의 복잡한 부분은 작은 공구를 가지고 절삭함으로써 절삭성을 향상시킬 수 있게 된다.

3.1 공구와 최소 Full형 팔분체와의 관계

황삭에 사용되는 공구의 종류는 평형 엔드밀(flat end mill)과 볼 엔드밀(ball end mill)이 대부분이다. 특히 평형 엔드밀은 볼 엔드밀에 비해 절삭력이 뛰어나기 때문에 황삭에서 보다 많이 사용되고 있다. 이러한 이유로 본 연구에서는 평형 엔드밀을 사용하도록 하였다.

앞에서 공구를 팔분체에 대응시킨다고 하였는데, 이는 팔분체의 크기가 알려지면 그에 맞는 공구를 선정할 수 있다는 것을 의미한다. 주어진 팔분체에 공구의 기하학적 대응은 Fig. 7에 표시되어 있다. 즉, 팔분체의 가로, 세로의 길이 d는 반경 절삭깊이(radial depth of cut)가 되고, 높이 h는 축 절삭깊이(axial depth of cut)가 된다.

먼저 공구의 지름 D를 구하여야 하는데, 이는 공구의 지름 D와 반경 절삭깊이 d의 관계로부터 그 범위를 결정할 수 있다. Fig. 8에서 하나의 팔분체를 절삭하기 위한 공구의 범위를 보이고 있는데, Fig. 8(a)는 가능한 최소의 공구를 보이고 있

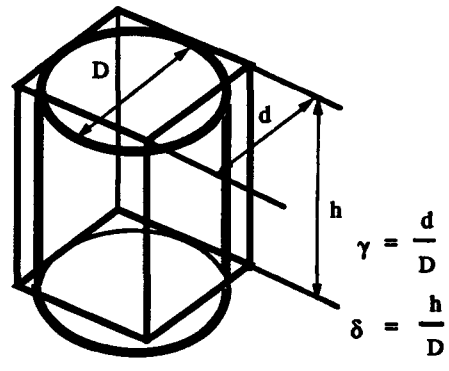
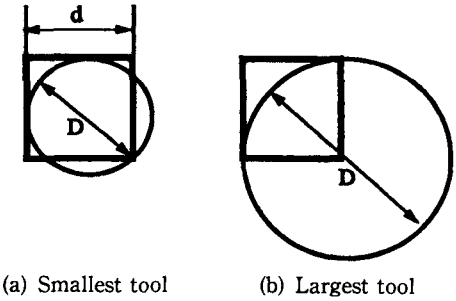


Fig. 7 Relation between tool and octant



(a) Smallest tool (b) Largest tool

Fig. 8 Possible range of tool size

고, (b)는 최대의 공구를 보이고 있다.

(a)의 경우에 D와 d의 관계는 다음과 같이 된다.

$$d=\frac{D}{2}+\frac{D}{2}\cos 45^\circ$$

$$\therefore D=\frac{1}{1+\cos 45^\circ}d=1.17d$$

또 (b)에서는 $D=2d$가 되므로 가로, 세로의 길이가 d인 팔분체를 가공할 수 있는 공구지름 D의 범위는

$$1.17d\leq D\leq 2d$$

가 된다. 따라서, 주어진 팔분체의 크기 d로부터 위의 조건을 만족하는 공구를 공구 데이터베이스로부터 찾으면 된다. 다시 말해 d/D를 γ라 하면,

$$0.5\leq\gamma\leq 0.95$$

를 만족시키는 공구를 선택하면 된다.

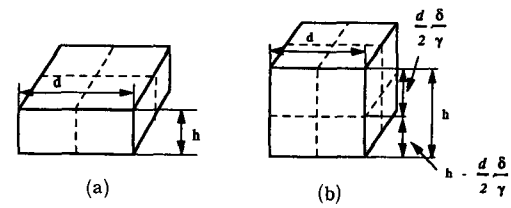
다음, 가능한 축 절삭깊이 h는 보통 공구지름 D의 1.5~3배 가량 되는데, 이들은 공구의 재료와 밀접한 관계를 가지고 있다. 즉, 공구의 재료가 고속도강일 경우 $h=1.5D$가 되고 공구가 brazed coated 엔드밀이면 h는 공구의 유효절삭길이 이내이면 된다.[16] 본 연구에서는 공구지름 D와 축 절삭깊이 h간의 h/D로 정하고, 이를 δ로 놓고 사용자가 그 값을 정하도록 하였다. 본 연구에서는 전체가공을 하는 동안 공구의 재질이 일정하다고 가정하여, δ를 항상 일정한 값을 갖도록 하였으며, 앞에서 팔분체의 가로, 세로의 길이를 고려하여 선택한 공구 중 공구지름이 h/δ보다 큰 공구가 대응되도록 하였다.

3.2 옥트리 생성과 처음 절삭공구 선택

만일 모델의 크기가 공구들에 비해 매우 크다고 하면, 처음 옥트리 분할을 하여 Full형 팔분체가 생성되더라도 생성된 팔분체에 맞는 공구를 선정할 수 없게 된다. 따라서, 분할을 계속하다가 대응되는 공구를 처음으로 찾을 수 있게 되었을때 선택되는 공구가 최초 절삭공구가 되어야 하는 데, 이는 최초에 옥트리 분할을 위해 만들었던 최소·최대 육면체의 크기에 따라서 달라지게 된다. 하지만, 최대의 절삭효율을 얻기 위해서 가지고 있는 공구 중에서 되도록 큰 공구가 최초 절삭공구로 선택되게 할 필요가 있다. 따라서, 본 연구에서는 이를 위한 알고리즘을 개발하였는데, 이는 최초 절삭공구를 먼저 정하고, 이에 맞는 팔분체가 생성되도록 최소·최대 육면체의 크기를 조정하는 방법이다. 이의 자세한 과정은 다음과 같다.

(1) 우선 공구 데이터베이스로부터 읽어들인 공구를 지름의 크기 순으로 정리한다.

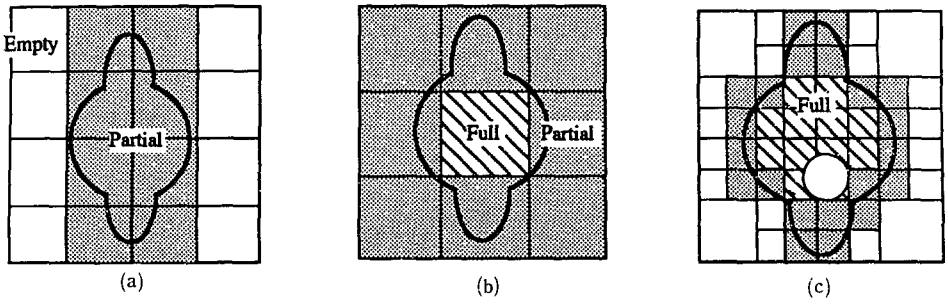
(2) 가장 큰 공구의 지름을 D라 하고 옥트리 분할의 초기 형상인 최소·최대 육면체의 가로, 세로 크기를 $D\cdot\gamma$의 2^n배로 하여 가공되어야 할 체적을 간신히 포함하도록 n의 최소 값을 찾는다. 2^n배가 되게 한 이유는 분할이 항상 1/2로 행해지기 때문이다. 여기서 γ는 앞의 3.1절에서 정의한 d/D인데, 최초 절삭공구에서는 최대의 반경 절삭깊이로 가공이 되도록 최대값인 0.85로 하였다. 한편 최소·최대 육면체의 높이는 절삭모델의 높이를 간신히 포함할 정도로 잡고, 옥트리 분할을 최소 팔분체의 가로, 세로가 $D\cdot\gamma$가 될 때까지 수행한다. 앞에서도 설명했듯이 본 연구에서는 높이가 절삭가능한 길이보다 작은 팔분체가 분할될 때는 높이 방향으로 분할될 필요가 없으므로 Fig. 9(a)와 같은 경우는 가로, 세로 부분만이 이등분 되어진다. 즉, 분할 되어야할 팔분체의 높이 h가 $\frac{d}{2}\cdot\frac{\delta}{\gamma}$보다 작



(a) $h<\frac{d}{2}\frac{\delta}{\gamma}$ Subdivision for 2 direction

(b) $h\geq\frac{d}{2}\frac{\delta}{\gamma}$ Subdivision for 3 direction

Fig. 9 Two types of subdividing



(a) Case where no full octant exists
(b) Case with a full octant but without a space for tool access
(c) Case with a space for tool access

Fig. 10 Three cases where octree should be divided

으면 가로, 세로방향으로만 분할이 되어진다. 그렇지 않은 경우는 Fig. 9(b)와 같이 높이 방향으로의 분할이 일어나게 되나 단순히 이등분되지 않고, 분할되어 생성된 8개의 팔분체들 중 위의 팔분체의 높이는 $\frac{d}{2}\cdot\frac{\delta}{\gamma}$가 되고 아래의 팔분체는 $h-\frac{d}{2}\cdot\frac{\delta}{\gamma}$가 된다. 이는 팔분체의 높이가 축 절삭깊이가 되므로 축방향으로 절삭을 가능한 한 최대로 하면서 공구를 사용하기 위함이다. 이것은 또한 팔분체의 아래면이 항상 절삭면이 되게 한다.

(3) (2)에서 분할된 옥트리에서 Full형 팔분체가 있는지 검사한다. 만일 Full형 팔분체가 없다면 가공되어야 할 체적이 작거나 복잡하여 현재의 절삭공구(공구지름 D)가 적합치 않은 것이므로, 그 다음으로 지름이 작은 공구를 가지고 다시 (2)의 과정을 반복한다. 예를들어 Fig. 10(a)와 같이 Full형 팔분체가 없을 때는 다음 크기의 공구로 바꾸어 (2)의 과정을 다시 수행해야 한다. $D\cdot\gamma$의 크기를 갖는 Full형 팔분체가 생기더라도, Fig. 10(b)처럼 공구가 진입할 공간을 제공할 수 있도록 충분한 개수의 Full형 팔분체가 존재하지 않는 경우엔 다시 더 작은 지름을 갖는 공구를 선택하여 (2)의 과정을 다시 수행한다. 형성된 Full형 팔분체들로부터 공구가 진입할 공간이 허용되는지의 판단방법은 3.3절에서 설명하겠다.

(4) Fig. 10(c)와 같이 현재의 공구로 절삭이 가능한 Full형 팔분체가 생기면 최초 절삭공구가 결정되고 따라서 이들을 절삭하는 공구경로를 생성한다.

위의 과정을 거쳐 공구를 선정하는 방법은 첫번째 절삭을 위한 최초 절삭공구를 찾을때 뿐이고, 두번째 절삭부터는 다시 분할된 팔분체로부터 3.1절의 조건을 만족하는 공구를 공구 데이터베이스에서 찾게 된다.

3.3 Full형 팔분체의 절삭

Fig. 11와 같이 Full형 팔분체가 생성되면, 이제 Full형 팔분체의 절삭 순서를 정하여야 한다. 그런데, Fig. 11에서 보는 것처럼 현재의 공구가 축방향으로 들어갈 수 있는 깊이는 h가 되므로 여러개의 층이 생길 수 있다. 또한 h는 3.1절에서 언급한대로 최소 Full형 팔분체의 높이이므로, $z=z_i$인 절삭면(cutting plane)은 최소 Full형 팔분체의 밑면이 된다. 각각의 팔분체의 높이는 가장 아래에 위치하는 팔분체를 제외하면 모두 h이므로, z_i는 다음과 같이 구할 수 있다.

$$z_i=z_{i-1}-h\,(i=1\cdots n-1)$$
$$z_0=z_{max}$$

이때, n은 층의 총 개수이며, z_{max}는 절삭모델의 최대 z값이다. 마지막 층을 차지하는 팔분체의 높이는 3.2절에서 언급한대로 h보다 작을 수 있으므로

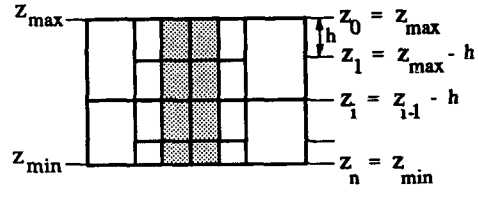


Fig. 11 Definition of cutting plane

$$z_n = z_{min}$$

이 된다. 이때 z_{min}은 절삭모델의 최소 z값이다.

다음은 Full형 팔분체들을 절삭하는 과정들이다.

(1) 제일 위의 층에서부터 각각의 층에 대해 다음 (2), (3), (4), (5)의 과정을 거친다.

(2) 현재의 층을 이루는 팔분체들을 모은다. Full형 팔분체의 크기는 여러가지가 있을 수 있으므로 이에 관계없이 절삭할 모든 Full형 팔분체를 구해야 한다. 이것은 다음의 조건을 만족하는 팔분체를 찾음으로써 이루어진다. 즉, Fig. 12처럼 현재의 가공해야할 층을 i라 하면 절삭평면이 $z=z_i$가 되므로 가공대상공간은 z_i와 z_{i-1} 사이에 위치하게 된다. 그리고, 임의의 팔분체의 최대, 최소 z값을 각각 o_{max}, o_{min}이라 하면, Full형 팔분체 중

$$o_{min} < \frac{z_i + z_{i-1}}{2} < o_{max}$$

을 만족하는 팔분체를 찾으면 된다. 위의 조건에 의하면 최소 Full형 팔분체보다 큰 팔분체들도 찾아지게 된다.

(3) (2)에서 구한 팔분체들의 z방향을 무시하면, Fig. 10의 (c)와 같은 쿼드트리가 된다. 따라서 한 층 내에서의 절삭은 이 쿼드트리에서 Full형의 사각형들에 대한 제일 바깥경계를 구하여 이 경계로부터 안으로 옵셋한 경로를 따라 행해진다.

(4) (3)에서 구한 경계를 옵셋한다. 처음 옵셋거리는 공구가 Full형 팔분체만을 가공해야 하므로 공구의 반지름만큼이 되어야 하며, 그 다음부터는 $0.85D$만큼 옵셋하게 된다. 이때 $0.85D$라는 옵셋거리는 Fig. 13[17]으로부터 정해지는데, (a)에서와 같이 처음에 반지름만큼 옵셋하여 절삭하면, 안쪽방향으로도 반지름만큼 절삭되어지므로, 그 다음 옵셋거리는 공구의 지름만큼 크게 할 수 있다. 그러나, 공구지름만큼 옵셋하게 되면, (a)에서처럼 절삭되지 않는 부분(island)이 발생하게 된다. 이 부분을 제거하기 위해 옵셋거리를 줄여야 하는데, 이를 위한 최대 옵셋거리는 Fig. 13으로부터 아래와 같이 된다.

$$L = \frac{1+\sin 45^\circ}{2} D = 0.85D$$

위 식은 경계곡선이 항상 90°로 만나므로 성립됨을 알 수 있다.

옵셋은 경계선분을 그것에 직각되는 양 방향 중 어느 한 방향으로 옵셋거리만큼 옮기면 쉽게 구할

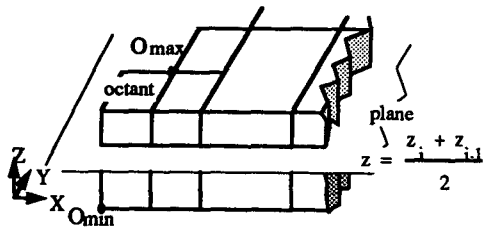
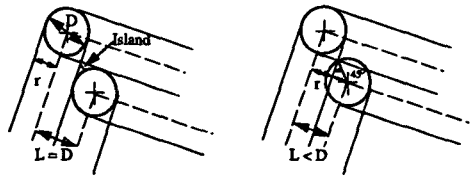
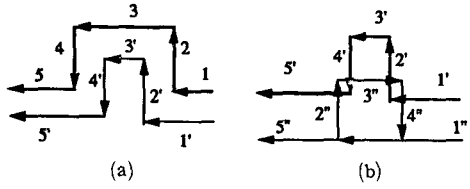


Fig. 12 Condition of intersection between plane and octant



(a) Case where offset distance is D
(b) Largest allowable offset to avoid island

Fig. 13 Decision of offset distance



(a) Boundary offset without self intersection
(b) Boundary offset with self intersection

Fig. 14 Two cases of boundary offset

수 있다. Fig. 14(a)는 한번 옵셋한 것을 보이고 있는데, 이 경우는 원래의 형상과 비슷한 형상이 생성되었다. 반면 Fig. 14(b)는 한번 더 옵셋한 것을 보이고 있는데, 이때는 선분 2″와 4″가 서로 자리를 바꾸면서, 3″의 방향이 바뀌게 된다. 이러한 결과는 공구의 직경이 가공형상에 비해 너무커서 가공형상을 제대로 가공할 수 없기 때문에 생긴 것이며, 이는 원하는 절삭을 할 수 없다는 것을 의미한다. 따라서, 이러한 Full형 팔분체의 경계를 옵셋한 전 옵셋곡선에 대해서 자기교차가 발생하면, 3.2절에서 기술한대로 Full형 팔분체가 존재하더라도 절삭할 팔분체가 없는 경우가 되어 현재의 공구선택이 취소되고 다시 새로운 옥트리가 생성된다. 그러나, Fig. 14(b)에서처럼 단지 국부적으로만 자기교차가 발생되면 자기교차가 발생한 부분에 해당

하는 팔분체만 현재 공구로 가공할 수 없다는 것이므로, 이러한 팔분체는 가공되지 않으며 이 곳을 지나게 하는 2", 3", 4"의 경로는 제거되고 단지 1", 5"의 경로로 가공하게 된다. 그리고, 여기에서 주의할 사항은 두번째 이후의 옵셋은 공구의 반지름보다 크게 되므로, 위와 같이 옵셋하여 자기교차가 생길지라도 Fig. 15(a)와 같이 가공이 가능한 경우가 생길 수 있다. 따라서 자기교차가 일어나 지워지는 선분들은 옵셋거리를 공구의 반지름만큼 해보아 Fig. 15(b)와 같이 다시 한번 더 검사해야 한다. 이때 다시 자기교차가 생기면 공구경로는 처음과 같아 지나, 자기교차가 생기지 않을 경우에는 처음 공구경로에서 공구의 반지름만큼 옵셋해서 구한 결과를 덧붙여야 한다.

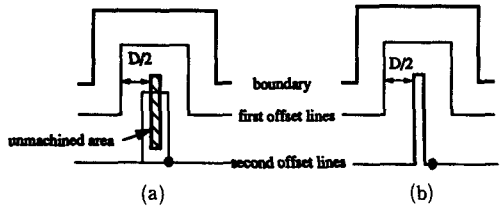
(5) 공구경로는 안에서 밖으로 향하게 되므로, 옵셋한 순서의 반대로 공구경로가 생성된다.

위에서 열거한 순서는 처음 선택한 최초 절삭공구에 대해 공구경로를 생성한 것으로, Fig. 16에서와 같이 이미 가공된 영역과 인접한 부분에서 공구경로를 구할 때는 안쪽의 부분은 이미 처음의 절삭에서 제거되었으므로, 옵셋하는 경계선분은 Partial, Empty 팔분체와의 경계만으로 이루어지게 하고, 이미 절삭된 팔분체와의 경계는 옵셋을 하지 않는다. 이렇게 옵셋하면 옵셋된 선분은 닫혀져 있지 않고 열려져 있게 된다. 이 열려진 양 끝점은 절삭되어 없어진 팔분체의 경계와 교점을 구해(Fig. 16에서 점 1,6) 그 교점을 양 끝점으로 치환한다.

3.4 분할의 끝 조건과 경계 프로파일링

3.3절에서 Full형 팔분체를 절삭하는 공구경로를 생성하는 방법을 설명하였는데, 한번의 절삭이 끝나면 남은 부분을 다시 한번 더 분할하여 새로 생긴 더 작은 Full형 팔분체에 대해 위의 과정을 반복하게 된다. 그러나 공구가 전체곡면을 가공할 수 있을 만큼 충분히 작아지면 옥트리 분할을 하지 않고, 경계를 프로파일링하게 하는 것이 좋다.

따라서, 더 이상 옥트리 분할을 해야 하는지 아닌지를 판단해야 하며, 그 방법은 다음과 같다. 우선 모델의 가공곡면과 $z=z_i$인 평면과의 교선을 구한다. 이때 z_i는 앞의 3.3절에서 팔분체를 구할 때 사용한 z_i와 같다. 교선을 구한 결과가 Fig. 17(a)와 같은 경우라면 현재의 공구가 전 곡면을 가공할 수 없으므로 프로파일링을 하지 않고 옥트리의 분할을 수행하며, Fig. 17(b)와 같이 되면 현재의 공구를 가지고 옥트리의 분할없이 경계부분을 프로파일링 할 수 있다.



(a) Unmachined area exists in spite of self intersection
(b) Modification of offset lines

Fig. 15 Problem caused by using offset distance bigger than tool radius

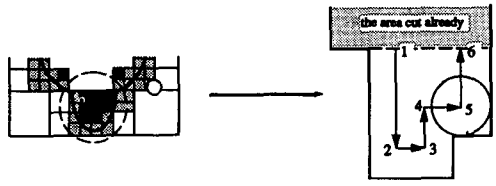
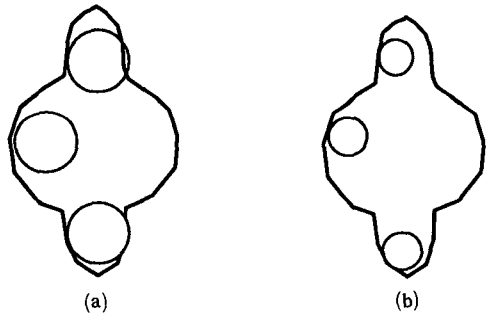


Fig. 16 Offsetting of open boundary



(a) Case where more subdivision is required
(b) Case where no more subdivision is required

Fig. 17 Condition for terminating subdivision

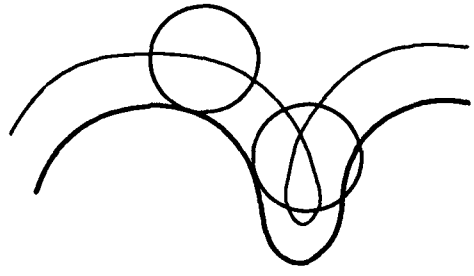


Fig. 18 Interference detection from self-intersection

위 두 경우의 판단은 간섭의 발생여부를 검사함으로써 이루어진다. Fig. 18에서는 공구중심의 궤적이 자기교차를 일으키게 되므로, 본 연구에서는 위에서 각 z_i에 대해 구한 교선을 현재의 공구 반지름만큼 옵셋하여 생성된 곡선이 자기교차가 발생하지 않게 되면, 더 이상 분할하지 않고 현재의 공구로 프로파일링하도록 하였다.

3.5 순서도

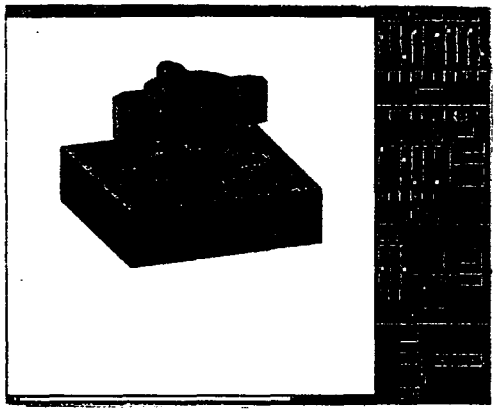
지금까지 설명한 공구경로의 생성과정을 분명히 하기 위해 다음과 같이 실제의 알고리즘을 가상언어로 표현하였다.

```
Rough_cutting ()
{
    INPUT () {
        Workpiece body model
        Part body model
    }
    PRE_PROCESS () {
        Get the cut body by Boolean operation and its minmax box
        Convert boundary faces to triangular facets
        Loading tools from the tool DB and sorting them
            in the descending order of their diameters
    }
    FOR (each tool) {
        Create root octant corresponding to the tool
        Generate octree
        IF (there is no full octant OR there is no machinable full octant)
            Next largest tool is selected;
        ELSE (Current tool is selected as the first tool to be used)
            BREAK;
    }
    FIRST_CUTTING () {
        Get the z value of layer from the height of the minimum octant
        FOR (each z value or each layer) {
            Get full octants
            Set cut flag of the full octants involved
            Get the boundary of full octants
            Offset the boundary until self-intersection is detected
            Tool path is generated with reversing the offset curves generated
        }
    }
    WHILE (TRUE) {
        CHECK_STOPPING_SUBDIVISION () {
            FOR (each z value) {
                Get the boundary from the intersection between model and
                cutting plane of z value
                Offset boundary
                IF (self-intersection)
```

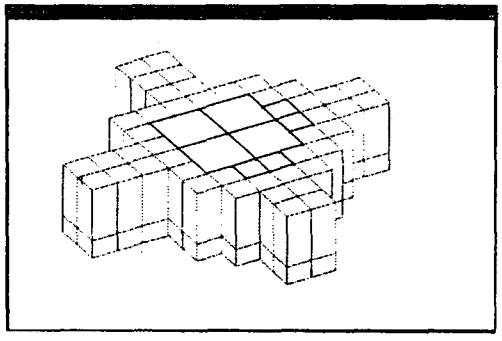
```
                    BREAK;
                }
                IF (NOT self-intersection)
                    BREAK;
            }
            SPLIT_ONE_MORE () { }
            MID_CUTTING () {
                Get full octants
                Set cut flag of the full octants involved
                Get boundary of full octants
                Offset boundary
                Tool path generation with reversing offsetting
            }
        }/*End of while clause*/
        PROFILING () { }
    }/*End of Rough machining*/
```

3.6 적용 예

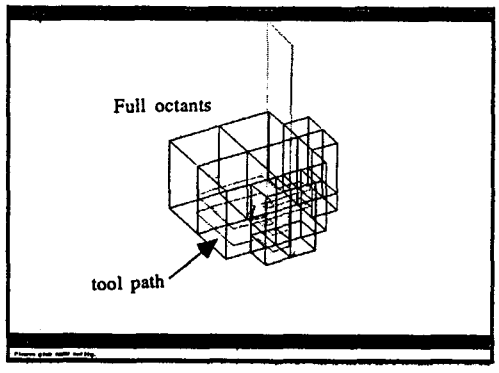
Fig. 19와 Fig. 20은 본 연구에서 개발한 알고리즘을 실제의 모델에 적용시킨 결과이다. Fig. 19(a)는 제품과 소재의 솔리드모델을 입력받아 제거되어야 할 형상을 나타낸 것으로 위의 모델이 절삭해서 제거해야 할 체적이 된다. (b)는 절삭모델에 옥트리를 생성시킨 것으로 진한 실선부분이 Full형이고 이 부분이 처음으로 절삭될 부분이다. 그림에서 연한 실선부분은 Partial형이다. Fig. 19(c)는 앞의 Full형 팔분체를 절삭하기 위한 공구경로를 보인 것이고, Fig. 19(d)는 다시 남은 Partial형 팔분체를 분할한 것으로 역시 진한 실선부분이 Full형이다. Fig. 19(e)는 마지막으로 프로파일링하기



(a) Solid model of the product and the volume to be machined

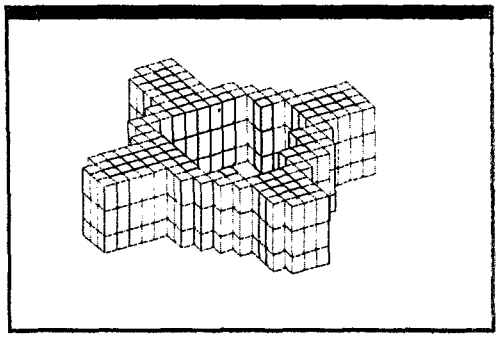


(b) Octree generation for selecting the first largest tool

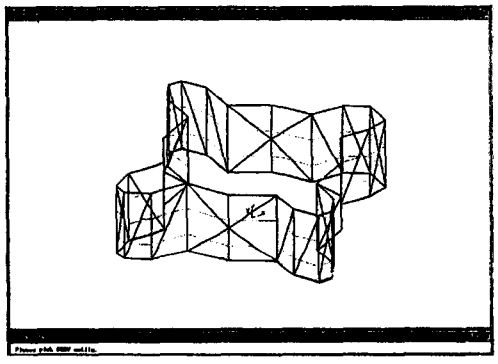


(c) Tool path of the first tool

Fig. 19 Result of rough cutting algorithm(continued)

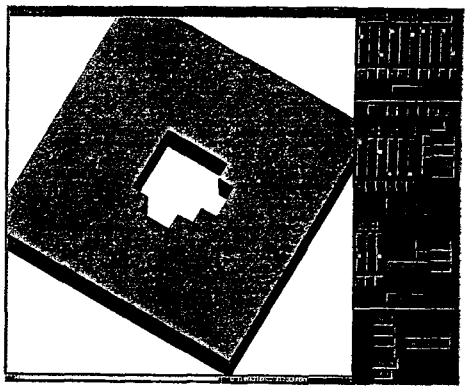


(d) Result after one more subdivision

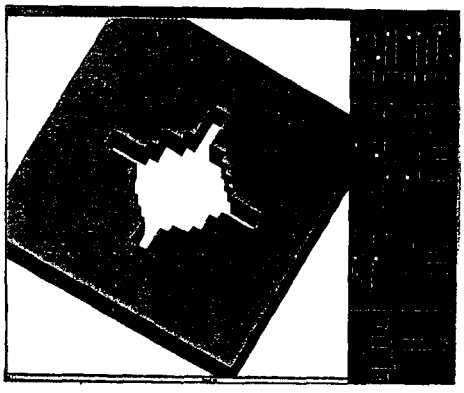


(e) Intersection lines for profiling

Fig. 19 Result of rough cutting algorithm



(a) Shape after first cutting



(b) Shape after second cutting

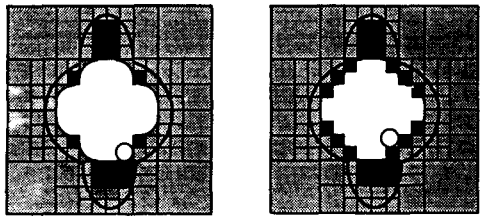
Fig. 20 Shaded image after each cutting

위해 절삭모델과 절삭면과의 교선을 구한 것이다. Fig. 19(e)에서 볼 수 있듯이 본 연구에서는 계산의 효율을 높이기 위해 절삭되어야 할 모델의 표면을 삼각형(facet)의 집합으로 근사하였다. 그리고 Fig. 20은 첫번째와 두번째 가공 후의 결과를 음영도로 나타낸 것이다.

4. 결 론

본 연구에서는 금형가공 등에 있어서 많은 시간을 소요하는 황삭가공을 위한 공구선정 및 공구경로 생성 알고리즘을 개발하였다. 이는 큰 공구로 많은 부분을 절삭한 다음, 작은 공구로 남아 있는 복잡한 부분을 절삭하도록 하여 효율적인 절삭을 목표로 하였다. 또한, 큰 공구에서 작은 공구경로 생성에 이르는 공구의 선택은 사용자의 입력에 의해서가 아니라, 시스템이 직접 기하학적인 정보를 이용하여 자동으로 선택하도록 하였다. 위 두가지 목표는 옥트리기법을 이용하여 만족시켰는데, 전자의 경우는 옥트리 분할을 할 때 생기는 큰 팔분체와 작은 팔분체를 순서대로 가공함으로써 구현하였고, 후자는 각각의 가공할 팔분체에 대응되는 공구를 구하는 알고리즘을 제시함으로써 가능하게 되었다.

그런데 팔분체를 실린더 형상의 공구에 대응시키면서 한가지 문제점이 발생하게 된다. 앞에서 Fig. 6(b)는 첫번째 절삭에서 Full형 팔분체를 제거한 후의 형상인데, 실제로 모서리 부분은 Fig. 21(a)와 같이 완전히 제거되지 않고 남아 있게 되므로 다시 옥트리 분할을 해서 얻은 Full형 팔분체를 가공할 때 이러한 미절삭 부분을 고려하지 않으면 공구파손 등의 예기치 못한 결과를 유발하게 될 것이다. 이것의 해결책으로 첫번째 절삭에서 제거되는 Full형 팔분체들 중에서 모서리 부분의 팔분체들을 Fig. 21(b)와 같이 이어지는 옥트리 분할시에 함께 분할하고 그 중에서 문제가 되는 팔분체를 Full형



(a) (b)

(a) Case where undercut is ignored
(b) Case where undercut is considered

Fig. 21 Undercut caused by matching cylindrical tool with octant

으로 전환해 주면 될 것이다.

그리고, 본 연구에서는 최소의 절삭시간이 걸리게 하기 위해 가능한 최대의 공구를 사용하였는데, 절삭시 소요되는 시간은 공구의 크기 뿐만 아니라 이송량과도 밀접한 관계가 있다. 즉, 공구의 크기가 증가하면 이송량은 줄어들기 때문에 이들의 적절한 배합을 취하게 하여 절삭시간을 최소로 할 수 있다. 따라서, 추후 연구과제로 공구의 크기와 이송량과의 관계를 고려하여, 절삭시간을 최소로하는 공구가 자동으로 선택되도록 하는 연구가 수행되어야 할 것이다.

참고문헌

(1) Suh, Y. S. and Lee, K. 1990, "NC Milling Tool Path Generation for Arbitrary Pockets Defined by Sculptured Surfaces," Computer Aided Design, Vol. 22, No. 5, pp. 273~284.

(2) 최재석, 1989, "NC 밀링가공시스템 개발에 관한 연구," 서울대학교 석사학위논문.

(3) Robert B. Jerard, et. al., 1989, "Methods for Detecting Errors in Numerically Controlled Machining Sculptured Surface." IEEE CG & A, Vol. 22, No. 1, pp. 26~39.

(4) Choi, B. K. and Jun, C. S., 1989, "Ball-End Cutter Interference Avoidance in NC Machining of Sculptured Surfaces," Computer Aided Design, Vol. 21, No. 6, pp. 371~378.

(5) Yoshimi, et. al., 1990, "5-Axis Control Machining Based on Solid Modeler," JSPE, No. 11, pp. 111~116.

(6) Takeuchi, Y. et. al., 1989, "Develoment of A Personal CAD/CAM Aystem for Mold Manufacture Based on Solid Modeling Techniques," Annals of the CIRP, Vol. 38, pp. 429~432.

(7) 1990, "Rough to Depth," Unigraphics II, Manufacturing Operation, Vol. 2, Chap. 26.

(8) 1991, "Strata™ Technical Overview," Spatial Technology Inc.

(9) Yamaguchi, K. et. al., 1984, "Computer Integrated Manufacturing of Surfaces Using Octree Encoding," IEEE CG & A, Vol. 17, No. 1, pp. 60~65.

(10) Yuen, M. F. et. al., 1987, "An Octree Approach to Rough Machining," Proc. Instn. Mech. Engrs., Vol. 201, No. B3, pp. 157~163.

(11) 이계정, 1991, "절삭성을 고려한 자유곡면 황삭 가공경로의 생성기법에 관한 연구," 서울대학교 석사학위논문.

(12) Lee, Yuan-Shin. et. al., 1991, "CASCAM-An Automated System for Sculptured Surface Cavity Machining," Computers in Industry, Vol. 16, pp. 321~342.

(13) Donald Hearn, et. al., 1986, "Computer Graphics," Prentice-Hall International

(14) Foley, et. al., 1989, "Computer Graphics Principles and Practice," 2nd edition, Addison Wesley.

(15) Jung, Y.H. and Lee, K., 1993, "Tetrahedron-based Octree Encoding for Automatic Mesh Generation," Computer Aided Design, Vol. 25, No. 3, pp. 141 ; 153.

(16) 1991, "'91 Performance Cutting Tools," Sumitomo Electric.

(17) Tlusty, J. et. al., 1990, "New NC Routines for Quality in Milling," Annals of the CIRP, Vol. 39/1, pp. 517~521.