

신속시작작업에서 2차원 단면데이터를 이용한 3차원 물체의 최적자동배치를 위한 알고리즘의 개발

허정훈*, 이건우**, 안재홍***

Optimal 3-D Packing using 2-D Slice Data for Multiple Parts Layout in Rapid Prototyping

Junghoon Hur, Kunwoo Lee, and Jaehong Ahn

ABSTRACT

In Rapid Prototyping process, the time required to build multiple prototype parts can be reduced by packing several parts optimally in a work volume. Interactive arrangement of the multiple parts is a tedious process and does not guarantee the optimal placement of all the parts. In this case, packing is a kind of 3-D nesting problem because parts are represented by STL files with 3-D information. 3-D nesting is well known to be a problem requiring an intense computation and an efficient algorithm to solve the problem is still under investigation. This paper proposes that packing 3-D parts can be simplified into a 2-D irregular polygon nesting problem by using the characteristic of rapid prototyping process that the process uses 2-dimensional slicing data of the parts and that slice of the STL parts are composed of polygons. Our algorithm uses no-fit-polygon (NFP) to place each slice without overlapping other slices in the same z-level. The allowable position of one part at a fixed orientation for given parts already packed can be determined by obtaining the union of all NFP's that are obtained from each slice of the part. Genetic algorithm is used to determine the order of parts to be placed and orientations of each part for the optimal packing. Optimal orientation of a part is determined while rotating it about the axis normal to the slice by finite angles and flipping upside down. This algorithm can be applied to any rapid prototyping process that does not need support structures.

Key words : Rapid prototyping, 3-D Packing, Slicing data, No-Fit-Polygon (NFP), Union of polygon, Optimization, Genetic algorithm

1. 서 론

공업 분야에서는 제품 개발 단계에서 시제품을 제작하여 이를 실제 분석, 부품 간의 조립성, 각 부품의 조립체 내에서의 기능성 등을 확인하거나 금형 제작등에 활용하는 것이 일반적이다. 그러나 이러한 시제품 제작과정은 설계의 초기 단계에서는 그 형상 자체가 매우 다양하고 소량이므로 매우 비효율적이라는 문제가 있으며 특히 제품의 수명주기가 짧아짐

에 따라 시제품 제작 시간도 단축되어야 하는 어려움이 있다. 신속 시작 작업(Rapid Prototyping, 이하 RP)은 이러한 문제점들을 해결할 수 있는 새로운 방식의 제품 제작 기법으로 최근 각광받고 있다.

RP란 짧은 시간 내에 3차원 CAD데이터로부터 바로 3차원 형상의 시제품을 만들어 내는 기술을 일컬으며, 1986년에 3D systems라는 회사에 의해 상용화된 SLA(Stereolithography Apparatus) 방식을 시작으로 지난 10년간 급속한 발전을 해왔다. 원하는 형상의 부품을 만들 때 기존의 절삭 가공은 원자재에서 재료를 깎아내면서 제작하는 반면에, RP는 Fig. 1에서와 같이 재료를 한 층씩 적층시켜 나가면서 제품

*학생회원, 서울대학교 기계설계학과

**중신회원, 서울대학교 기계설계학과

***한국과학기술연구원

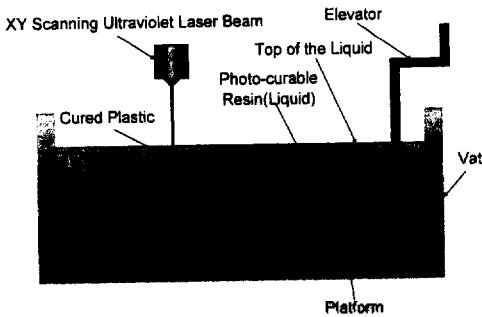


Fig. 1. Methodology of stereolithography apparatus.

을 만드는 공정 특징을 가지고 있다. RP 가공은 NC와 같은 절삭 가공이 한번의 셋업(setup)으로 가공할 수 없는 형상도 한 번에 가공할 수 있으며 절삭 가공에 비해 가공 시간이 짧아 전체 제품생산시간을 줄일 수 있고 생산 비용도 절감할 수 있는 장점이 있다. 특히 시작품과 같이 제작 시간이 짧고 소량 생산이어야 하는 경우 좋은 결과를 얻을 수 있다¹¹⁾.

RP 가공은 전술한 장점 외에도, 여러 개의 제품을 한 번의 성형으로 제작할 수 있다는 장점을 가진다. 즉 단면 정보를 구한 여러 개의 모델을 작업 공간 내에 중첩되지 않게 배치한 후 가공을 할 경우에 제품 사이의 빈 공간을 최소화하여 제품들이 차지하는 높이를 줄이게 되면 전체 가공시간을 단축하는 효과를 가져올 수 있게 된다. 이는 층별가공을 하는 RP 가공상의 특성상 같은 층에서 성형에 소요되는 시간보다는 층 간을 이동하는 시간이 전체 성형시간에서 차지하는 비율이 높기 때문이다. 또한 전체 성형 횟수를 줄임으로써 한 번의 성형이 끝난 뒤 다음 성형을 위해 준비하는 시간과 사용자의 수고를 덜 수 있으며 RP 가공방식에 따라서는 재료를 절감하게 해주거나 레이저의 수명을 연장시켜 주는 효과를 주기도 한다¹⁾.

RP 시스템에서 이러한 패키징 기능을 지원하지 않으면 사용자가 수작업으로 패키징을 해야 하는데 이 경우 정밀도가 떨어지고 패키징을 할 때에 각 모델이 가질 수 있는 변수가 적어질 수 밖에 없으며 3차원의 공간에서 이루어지기 때문에 작업 자체가 매우 어려운 단점이 있다. 또한 이와 같은 패키징 문제는 NP(Non-deterministic Polynomial time) complete의 특성을 가지므로 최적의 해를 찾는 것이 어려우며 이를 사용자의 수작업으로 최적화된다는 것은 거의

불가능할 수 밖에 없다. 따라서 여러 변수값들을 최적화하여 자동으로 모델들을 작업 공간 내에 패키징하는 것은 성형에 드는 시간을 줄일 수 있을 뿐만 아니라 사용자의 수고를 덜 수 있으며 수작업에 의한 결과보다 좋은 결과를 가져올 수 있다는 이점이 있다.

본 연구에서는 RP 가공에서의 가공시간 단축을 위한 3차원 패키징 최적화를 수행하고자 하였으며, 특히 RP 프로세스가 성형 시에 물체의 2차원 단면 데이터²⁾를 기반으로 한다는 특성을 이용하여 성형하고자 하는 3차원 모델들의 위치를 2차원 상에서의 단면데이터들의 최적배치문제로 변환하여 문제에 접근하였다. RP 성형 방식은 재료와 성형 방법에 따라 여러 가지가 있으나 2차원 단면 정보를 이용하여 모델을 성형한다는 공통점이 있으므로 본 연구는 일반적인 RP 성형에 적용될 수 있다.

본 연구에서는 2차원 단면들 간의 중첩이 발생하지 않도록 각 물체를 배치하기 위해 No-Fit-Polygon의 개념을 응용하였고 최적화를 위해서는 조합최적화 문제에 적용이 용이한 유전 알고리즘(Genetic algorithm)을 사용하였다.

2. 관련연구

RP에서는 각 모델들이 3차원 형상으로 제작되므로 본 문제는 곧 제한된 공간 내에 가능한 한 많은 수의 모델을 배치하거나 일정 개수의 모델들이 차지하는 부피가 작도록 배치하는 3차원 패키징 문제로 요약될 수 있다. 패키징 문제는 2차원의 경우 많은 연구 결과들이 나와 있으나 3차원 패키징에 대해서는 응용 분야가 적고 다루어야 할 변수가 많아 구현이 어려운 이유 등으로 인하여 연구 결과가 2차원에 비해 적은 실정이다.

2차원에서의 패키징 문제는 주로 의류가공, 피혁가공, 선박제조, 항공산업, 박판 가공 등의 분야에서 재료 활용도를 극대화하여 재료의 낭비를 막기 위한 방안으로 활발히 연구되어 왔다.

Adamowicz 등은 No-Fit-Polygon을 이용하여 2차원의 형상들을 사각형 모듈로 만들어 네스팅(nesting)을 하는 방법을 제시하였다¹²⁾. 그들은 여러 형상들을 둘러싸는 직사각형을 No-Fit-Polygon의 개념을 이용하여 구하고 이를 모듈(module)이라 하였다. 구해진 여러 개의 모듈 중 최소의 면적을 갖는 모듈을 구하고 이러한 모듈들에 대해 최적의 레이아웃을 구하는

¹⁾SLA 가공에서는 매 2000~3000 시간마다 레이저를 교체해 주어야 한다.

²⁾SLI, SLC 등의 파일형식.

알고리즘을 제시하였다.

Ismail 등은 압연 공구의 설계를 위한 2차원 네스팅에 대한 연구 결과를 발표하였다^[3]. 그들이 제시한 알고리즘에서는 형상을 둘러싸는 직사각형의 격자로부터 0과 1로 이루어지는 격자 배열을 만들었다. 그리고 격자 배열의 각 변에 인접해 있는 0의 개수를 저장한 모서리 배열을 만들어 이를 통하여 형상 정보를 구함으로써 최적의 패어링(pairing)을 구하는데 이용하였다. 그들은 또한 격자를 이용한 2차원 네스팅 알고리즘을 최적화하기 위한 유전 알고리즘에 효율적인 유전 연산자를 제시하기도 하였다^[4].

조경호 등은 불규칙한 형상의 박판 위에 2차원 형상들을 네스팅하는 알고리즘을 개발하여 피혁가공 등에서 발생하는 실제적인 문제점에 적용할 수 있게 하였다^[4]. 그가 제시한 알고리즘은, 기존의 네스팅 알고리즘들이 직사각형 내부에서의 네스팅을 시도함으로써 가지게 되는 한계점을 극복하였고, 내부에 결함이 있는 경우 및 여러 시트(sheet)에 네스팅해야 하는 경우들을 고려하기도 하였다. 알고리즘의 구현을 위해 2차원 형상과 시트를 격자로 표현 하였고 시뮬레이터더어닐링(simulated annealing)을 이용해 최적화를 수행하였다.

이러한 2차원 패킹 문제에서는 대개 경험적(heuristic) 탐색 방법이나 확률적인 접근 방법이 사용되었는데, Ismail^[5], Jakobs^[6], Jain^[7] 등은 유전 알고리즘(genetic algorithm)을, Albano^[8] 등은 경험적 기법을 패킹 문제에 적용하였다.

3차원에서의 패킹 문제는 화물 적재, 상품 포장 등과 같은 분야에서 주로 연구 결과들이 나와 있으며 RP 분야에서도 필요성이 부각되어 여러 가지 패킹 알고리즘이 연구되고 있다. 3차원 형상의 패킹에 대해서는 여러 최적화 기법이 적용되었으나 모델의 형상이나 모델의 방향(orientation), 문제의 복잡성 등에 있어 제약이 많았다.

Wodziak 등은 SLA 가공을 고려한 3차원 패킹을 연구하였다^[9]. 그들이 제시한 알고리즘은 각 물체를 둘러싸는, 모서리가 x 또는 y축에 평행한 방향을 갖는 직육면체를 구한 다음 이들을 유전자알고리즘을 이용하여 최적화를 수행하였다. 따라서 패킹의 결과가 완전한 최적해가 아니라는 점과 물체 내부에 구멍이 존재하는 경우 이들의 형상이 무시되어 버리는 단점이 있다.

Magics^R RP는 벨기에의 Materialise사에서 제작한 RP 작업을 위한 STL 파일기반 소프트웨어로서 STL모델들의 패킹기능을 지원한다^[10]. 그러나 Mag-

ics^R RP의 모델 자동배치기능은 기본적으로 2차원 상에서 이루어지기 때문에 모델의 수가 많아서 z축 방향으로의 모델의 이동이 필요한 경우에 이를 수행하지 않고 경고메시지와 함께 실제 플랫폼의 바깥영역에 모델들을 배치하게 된다. 또한 모델을 배치시킬 때 물체를 둘러싸는 최소 크기의 직육면체를 사용하여 배치과정 중에 어떠한 축방향으로의 회전 또한 수행하지않는다. 따라서 자동 배치의 결과는 완전한 최적해라고 말할 수 없다.

3. 패킹수행을 위한 데이터의 준비

RP 가공을 이용해서 제작할 3차원 모델은 STL 파일 형식의 데이터로 변환되어야 한다. 대부분의 상용 CAD 시스템이 STL 파일 형식의 데이터를 지원하고 있으며 각종 CAD 데이터를 STL 파일 형식으로 변환해 주는 소프트웨어들도 많이 개발되어 있다. STL파일은 3-D Systems사를 위해 Albert Consulting Group에서 개발한 파일 형식으로서 오늘날 대부분의 RP 가공의 표준 데이터가 되었으며 이는 3차원의 단면진 모델을 삼각형 패치(facet)로 근사화한 것이다. STL 파일은 삼각형의 법선 방향 벡터와 삼각형의 세 꼭지점을 나타내는 x, y, z 좌표로 구성되어 있다. Fig. 2에 STL 파일 형식의 예를 나타내었다^[11].

STL 파일 형식으로 나타내어진 3차원 형상의 단면 정보(slicing data)는 성형 방향에 수직한 평면들과 STL 모델과의 교점을 구하여 구하게 되는데 STL 모델이 삼각형 패치의 집합으로 이루어져 있기 때문에 결과적으로 단면 정보는 여러 개의 선분들로 이루어진 다각형이 된다. 이러한 단면 정보를 저장하는 SLI 또는 SLC 등과 같은 파일 형식은 RP 가공 시스템에 따라 다르다.

본 논문에서 제시한 알고리즘에서는 3차원 모델들의 각각의 단면 정보가 모두 구해진 후 그 단면 정보를 이용해 패킹이 수행된다. 따라서 알고리즘을 수행하기 위해 필요한 형상 정보 파일은 STL파일이 아닌 단면정보를 담고있는 SLC 파일 형식을 적용하였다. 단면 정보 파일에는 모델의 이름, STL 파일명, 층 두께, 각 단면의 꼭지점의 좌표 등이 저장된다. 단면은 바닥면에서부터 저장되며 단면 내의 각 루프의 꼭지점의 x, y 좌표를 기록한다. Fig. 3에는 본 논문에서 사용한, Stratasys사의 FDM SLC 파일의 형식을 보여 주고 있다. 일반적으로 실제 RP 성형에 사용되는 단면정보 파일에는 매우 많은 수의 단면들

```

solid test_sphere
facet normal -0.13 -0.13 -0.98
  outer loop
    vertex 1.50000 1.50000 1.50000
    vertex 1.50000 1.11177 0.05111
    vertex 1.11177 1.50000 0.05111
  endloop
endfacet
...
endsolid test_sphere
    
```

(a) ASCII STL file

Byte	Type	Description
00	string	Head information such as the CAD system used
4	unsigned long integer	Number of facets
First Triangle Definition		
4	float	normal x
4	float	normal y
4	float	normal z
4	float	vertex1 x
4	float	vertex1 y
4	float	vertex1 z
4	float	vertex2 x
4	float	vertex2 y
4	float	vertex2 z
4	float	vertex3 x
4	float	vertex3 y
4	float	vertex3 z
2	unsigned integer	Number of attributes bytes should be set to zero
Second Triangle Definition		
4	float	normal x
4	float	normal y
4	float	normal z
4	float	vertex1 x
4	float	vertex1 y
4	float	vertex1 z
4	float	vertex2 x
4	float	vertex2 y
4	float	vertex2 z
4	float	vertex3 x
4	float	vertex3 y
4	float	vertex3 z
2	unsigned integer	Number of attributes bytes should be set to zero

(b) Binary STL file

Fig. 2. Example of STL file.

포함하고 있기 때문에 이들을 모두 최적화 수행에 사용한다는 것은 불가능하다. 따라서 본 논문에서는 이들 단면정보 데이터에서 일정 간격으로 몇 개의 단면 데이터들만 취하여 최적화를 수행하게 된다. 이는 물체의 형상을 표현하는데 있어서 대부분의 단면 데이터들은 실제로 불필요하다는 이유에서 기인한다.

단면 정보를 구할 때 모델의 방향(orientation)은 사용자나 시스템에 의해 정해진다. 이때 모델의 방향은 여러 가지 조건에 의해 정해질 수 있다. 이는 주로 가공방법에 따라 달라지는데, 특별한 제한 조건이 없는 가공방법도 있다. 예를 들면, SLA나 FDM과 같이 지지대(support)가 필요한 가공방법의 경우,

```

$lice V0          # 0 = zero
FILE( filename) # filename may include full path
SLICES( )        #
Z z dz          # z = absolute z of layer, dz = layer thickness
x y             # Provide the coordinates of the corner points
x y             # along each contour.
x y             # Do not duplicate the first point.
x y             # The is no upper limit on the number of points
x y             # in a contour.
C               # Terminate the contour with a "C".
x y             # Begin the second contour on this level.
x y             #
x y             #
x y             #
x y             #
x y             #
C               # Terminate the second contour.
Z z dz         # Begin a new layer (z = z.old + dz)
x y           # Begin a new contour on a new level.
x y           #
x y           #
x y           #
x y           #
x y           #
C             # Terminate the contour.
END           # All following lines are ignored.
    
```

Fig. 3. Example of SLC file.

지지대의 체적을 최소로 하여 재료를 절감하는 방향으로 모델의 방향을 잡을 수 있다. LOM과 같은 경우 단면의 면적이 너무 작으면 부러질 수 있으므로 모델의 형상에 따라 방향을 결정하기도 한다. 또는 가공 방법에 무관하게, 모델의 높이를 최소로 하여 가공시간을 줄이는 방향으로 모델을 두고 단면 정보를 구할 수도 있다. 이와 같이 x, y 축에 대해 회전 변환에 의해 결정되는 방향은 가공 방법의 종류나 사용자의 요구에 따라 달라지게 된다^[11]. 따라서 이러한 특성을 고려하여, x, y 축에 대한 모델의 회전량은 단면 정보를 구할 때 이미 결정되어지는 것으로 가정한다. 최적의 패킹 패턴(packing pattern)을 찾기 위해 가능한 모든 변수를 도입하게 되면 그 만큼 많은 시간이 소요되게 되는데, RP의 특성에 따르는 이러한 제한 조건을 적용함으로써 탐색 영역을 많이 줄일 수 있게 된다.

4. 단면데이터를 처리하기 위한 데이터 구조

SLC 파일로부터 단면 정보를 읽어 들여서 각 단면에 해당하는 데이터들을 Fig. 4에 나타난 자료 구조 내에 저장한다. Fig. 4는 단면의 정보를 저장하기 위한 자료 구조를 나타낸 것으로서 PointLIST는 각 꼭지점들을 리스트로 연결해 루프를 구성하기 위한 구조체로서 꼭지점의 좌표값을 저장하게 된다. SliceLIST는 한 단면 내에 있는 루프를 나타내기 위한 구조체로서 모델의 모든 루프는 리스트로 연결된다. 각 루프는 PointLIST를 포함하며 루프의 종류, z값 등을 가지고 있다. Part 구조체는 모델에 관련된 여

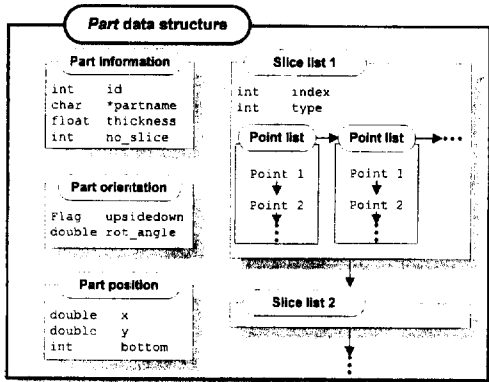


Fig. 4. Data structure for handling slice data.

러 정보값 및 패키징 과정에 필요한 변수들을 가지고 있다. 또한 모델의 단면 정보를 저장하기 위해 SliceLIST의 선두 루프를 가리키는 포인터를 가지고 있다.

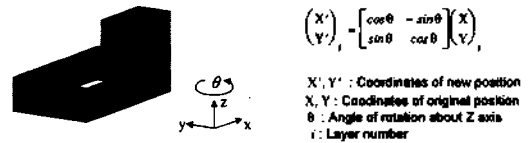
5. 패키징수행을 위한 방향변수의 설정

패키징을 수행하고자 하는 모델들의 단면 정보를 읽어 들어 자료 구조에 저장한 후에는 각 모델의 방향을 정해야 하는데 이는 z축에 대해 주어진 각도만큼 회전을 하거나 모델의 위 아래를 바꾸는 방식으로 이루어진다. 각 모델의 방향 변수(위 아래를 바꿀 것인지의 여부를 나타내는 값, z축에 대한 회전 각도) 및 모델들의 패키징 순서는 최적화 수행과정의 각 단계에서 정해지며 방향이 결정된 모델들은 주어진 패키징순서에 따라 정해진 패키징 알고리즘에 의해 최적의 위치가 결정된다.

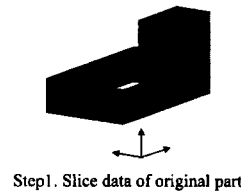
최적화 과정에서 각 모델에 대해 주어진 방향 변수에 따라 모델의 방향을 바꾸는 과정은 Fig. 5에서와 같이 모델의 2차원 단면 정보를 저장하고 있는 자료 구조내의 각 꼭지점들을 변환함으로써 이루어진다. z축에 대한 회전을 할 경우에는 변환 행렬을 이용하여 꼭지점들을 각 단면상에서 회전 이동시킨다. 모델의 위 아래를 바꿀 때에는 각 단면별로 꼭지점들을 2차원 상에서 x축(또는 y축)에 대해 180° 만큼 회전시킨 후 단면 리스트의 순서를 바꿔준다. 이것은 3차원 모델의 위 아래를 바꾼 후 단면 정보를 구한 것과 같은 결과가 된다.

6. 모델 자동배치 알고리즘

6.1 Bottom-MinY-MinX 조건



(a) Rotation about Z axis



(b) Flipping upside down

Fig. 5. Manipulation for part orientation.

방향이 결정된 각 모델의 위치를 주어진 순서에 따라 순차적으로 결정하는 자동배치 알고리즘은 모델들이 중첩되지 않으면서 전체 모델이 차지하는 층의 높이가 최소가 되도록 각 모델의 위치를 결정한다.

패키징 문제에서는 모델을 이동함에 따라 패키징 패턴이 모두 달라지게 되어 패키징 문제에 대한 탐색 영역은 무한히 커지게 된다. 이러한 문제점에 대해 패키징 패턴의 수를 효과적으로 줄이기 위한 방법의 하나로써, Bottom-Left 조건(BL 조건)이라는 것이 있다. 즉, 배치된 모델이 더 이상 아래쪽이나 왼쪽으로 이동하지 못하는 경우, 이때의 패키징 패턴은 BL 조건을 만족하는 것이다. 또한 이러한 조건을 만족하도록 모델을 배치하는 2차원 패키징 알고리즘을 BL 알고리즘이라 한다.

BL 알고리즘에 따라 패키징하고자 하는 n개의 모델을 다음과 같이 나타낸다고 하자.

- π : n개의 모델을 배치하는 순서
- $P_{\pi(i)}$: 주어진 순서 중 i번째에 해당하는 모델 ($i=1, 2, \dots, n$)

이때 모델들은 다음과 같은 과정에 따라 배치가

된다.

Step 1. $P_{\pi(i)}$ 을 2차원 작업 영역의 좌측 맨 아래에 배치한다.

Step i. 작업 영역의 우측 맨 위에서부터 시작하여 $P_{\pi(i)}$ 를 최대한 아래까지 이동한 후 다시 최대한 왼쪽으로 이동한다.

($i=2, 3, \dots, n$)

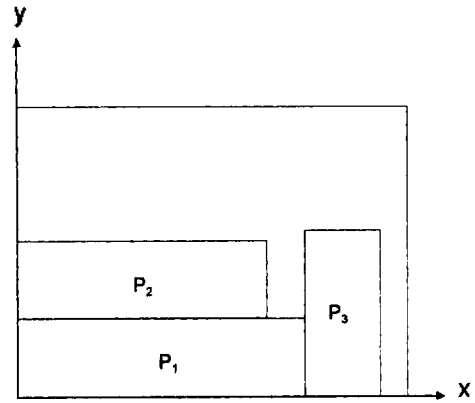
위와 같은 BL 알고리즘에 따라 배치된 패킹 패턴은 BL 조건을 만족하게 된다. BL 알고리즘에 의해 사각형들을 배치한 예를 Fig. 6에 나타내었다.

BL 알고리즘에 의해 생성되는 가능한 패킹 패턴의 수에는 상한이 존재한다. n 개의 모델이 주어지고 각 모델은 m 개의 방향을 가질 수 있다고 하면 BL 알고리즘에 의해 계산된 패킹 패턴이 가질 수 있는 수의 상한은 $m^n \cdot n!$ 이 된다. 그러나 실제로 BL 알고리즘에 의해 생성되는 패킹 패턴은 $m^n \cdot n!$ 보다 작다. Fig. 7을 보면, 배치 순서가 다른 두 경우에 대하여 같은 패킹 패턴이 생성됨을 알 수 있다.

이와 같은 BL 알고리즘과 유사하게, 본 논문에서 제시한 알고리즘에서는 3차원의 작업 공간 내에서 *Bottom-MinY-MinX*의 조건을 만족하도록 3차원 모델을 배치하도록 한다. 즉 모델의 위치를 결정할 때 좌표의 우선 순위가 Z-Y-X의 순서임을 의미한다. 즉 최소의 Z값을 결정한 후 최소의 Y, 최소의 X값을 찾아 모델의 위치를 정하는 것이다. 여기에서 Z 방향은 성형 방향을 의미하며 X, Y는 각각 작업 공간의 길이방향, 너비 방향을 의미한다.

3차원 작업 공간에 적용되는 좌표계와, *Bottom-MinY-MinX* 조건에 따라 모델을 배치하는 개략도를 Fig. 8에 도시하였다.

이러한 방법에 의해, 각각의 모델은 사용 가능한 공



Locating order : Case 1 = (P₁, P₂, P₃)

Case 2 = (P₁, P₃, P₂)

Fig. 7. Same result for the different packing order.

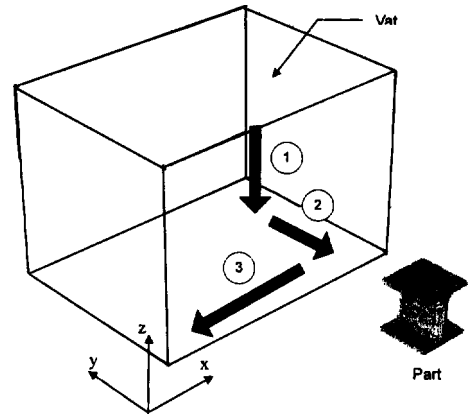


Fig. 8. Locate a part with *Bottom-MinY-MinX* condition.

간 내에서 가장 아래층에 놓이며 각 층에서는 2차원에서의 BL 알고리즘과 같은 방법으로 배치가 된다. 즉 다음과 같은 과정에 따라 순차적으로 배치가 된다.

Step 1. $P_{\pi(1)}$ 을 3차원 작업 영역의 맨 아래에서 최소의 x, y 값을 갖는 위치에 배치한다(Fig. 9(a)).

Step i. 작업 영역의 맨 아래에서부터 시작하여 $P_{\pi(i)}$ 이 놓일 수 있는 영역을 찾아 나간다. 사용 가능한 영역이 존재할 경우 그 높이에서 최소의 y값을 갖도록 모델을 y방향으로 이동한 후 다시 최소의 x값을 갖도록 x방향으로 이동한다(Fig. 9(b)).

($i=2, 3, \dots, n$)

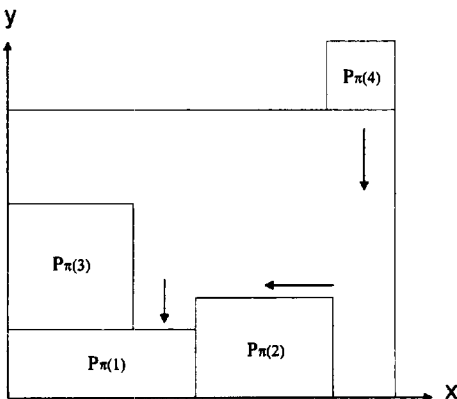
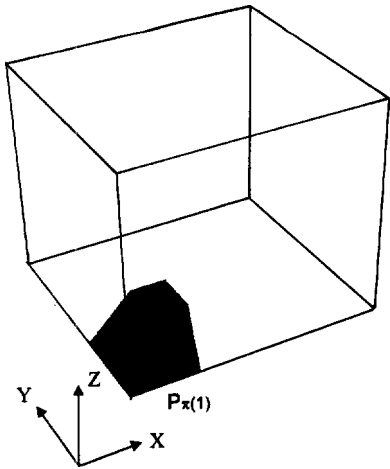
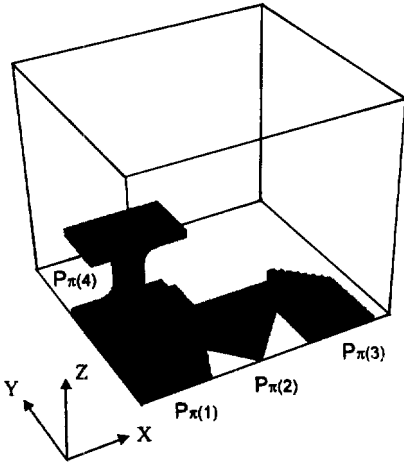


Fig. 6. An Example of BL-algorithm.



(a) Step i



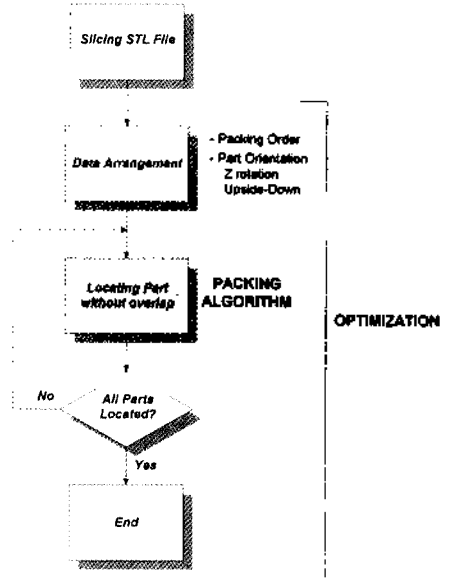
(b) Step i

Fig. 9. Procedure of three-dimensional packing process.

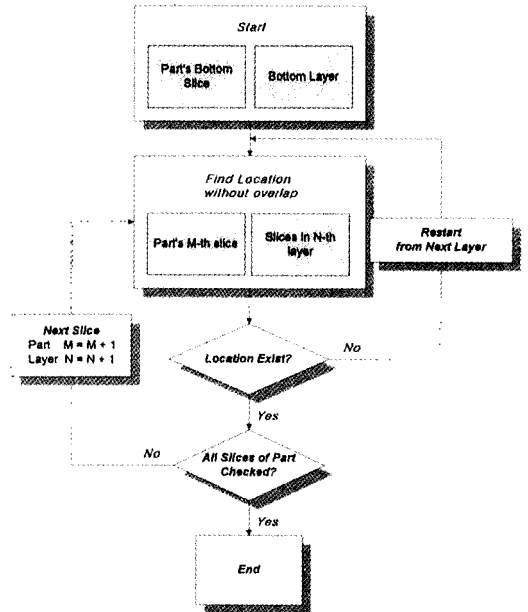
6.2 모델 간의 중첩검사

각 모델의 방향이 결정되고 배치 순서가 주어지고 나면 그 순서에 따라 각 모델이 놓일 수 있는 영역을 구한다. 사용 가능한 영역을 찾는 것은 앞서 배치된 모델들과 중첩이 되지 않는 영역을 찾는 것이라 할 수 있다. 모델 간에 중첩이 발생하지 않는 위치를 찾기 위한 중첩 검사(overlap check)는 최소의 Z, 즉 작업 영역의 맨 아래에서부터 이루어진다. 모델이 놓일 수 있는 영역이 구해지고 나면 *Bottom-MinY-MinX* 조건에 따라 그 영역 중에서 최소의 Y, 최소의 X값을 가지는 위치에 모델을 배치한다. 만약 배치 가능한 영역이 작업 공간 내에 존재하지 않을 경우

작업 영역의 다음 층에서 다시 중첩 검사를 한다. 즉 모델의 위치를 작업 영역의 바닥면에서부터 찾아 나가고, 모델이 놓일 수 있는 공간이 없으면 한 층 위에서 다시 위치를 찾는 것이다. 이와 같은 방법은,



(a) Packing process



(b) Packing algorithm

Fig. 10. Algorithm for optimal packing.

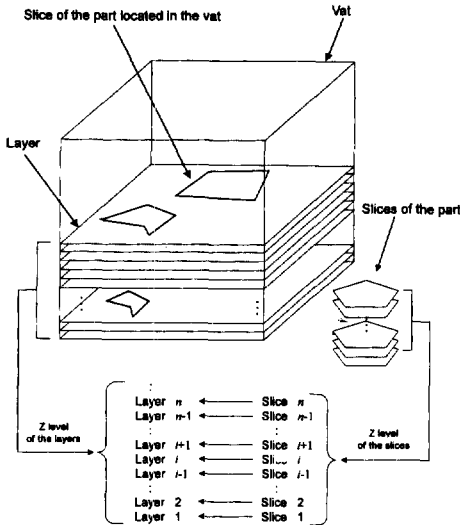


Fig. 11. Relationship between part slice and building layer.

작업 공간 내에 앞서 놓여진 모델들이 차지하지 않는 빈 공간 중에서 가장 아래층에 새로운 모델이 놓일 수 있도록 해 준다. 즉 *Bottom-MinY-MinX* 조건에 첫번째 조건을 우선적으로 만족하게 해 주는 것이다.

Fig. 10은 이 알고리즘을 순서대로 나타낸 것이다. 여기서 (a)는 최적화를 포함한 전체 과정을 나타낸 것이며 (b)는 전체 과정 중, 모델을 배치하는 패키징 알고리즘만을 나타낸 것이다.

중첩 검사는 단면별로 2차원 상에서 이루어진다. 이때 모델의 단면이 이루는 다각형과 앞서 놓여진 모델들의 단면이 이루는 다각형들간에 중첩 검사를 하게 된다. 각 단면은 대응하는 층에 대해서만 중첩 검사를 하면 되는데, 모델의 단면과 작업 공간내의 층과의 대응관계를 나타내면 다음의 Fig. 11와 같다.

결과적으로 모델의 각 단면이 작업 공간 내의 대응 층 상에서 놓일 수 있는 2차원의 영역을 본 알고리즘에 의해 찾게 되면, 3차원상에서 모델이 작업 공간 내에 놓일 수 있는 공간을 구할 수 있게 된다.

6.3 No-Fit-Polygon

본 알고리즘에서는 단면간에 중첩이 없는 위치를 결정하기 위한 중첩 검사를 하기 위해 No-Fit-Polygon(이하 NFP)의 개념을 사용하였다.

방향이 고정된 두 개의 다각형 A, B가 있을 때 A는 고정되어 있는 상태에서 A와 중첩되지 않고 놓일

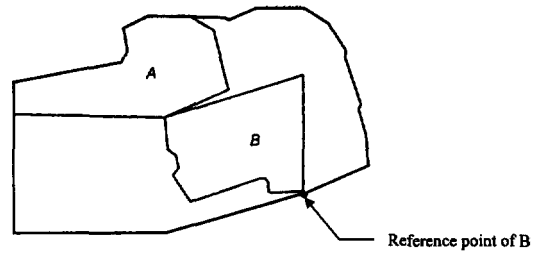


Fig. 12. No-Fit-Polygon.

수 있는 B의 위치를 구한다고 하자. 다각형 B가 다른 다각형 A에 중첩됨 없이 접촉하기만 하면서 움직이도록 할 때 B의 기준점이 그리는 자취를 NFP라 한다. NFP의 개념을 Fig. 12에 도시하였다.

B의 기준점이 NFP의 내부에 있을 때 다각형 A, B 사이에는 중첩이 발생하게 된다. 따라서 중첩이 없는 위치를 구하고자 하는 다각형 B가 다른 다각형 A와 이루게 되는 NFP를 구하게 되면 그 NFP의 바깥 영역에 B의 기준점을 놓음으로써 중첩이 없이 다각형 B를 배치할 수 있다. 이러한 성질을 이용하여 모델의 단면이 층에 배치된 단면들과 중첩하지 않는 영역을 구할 수 있다.

6.4 기준점 맵(Reference Point Map)

다각형 B의 기준점은 반드시 B의 모서리 위에 있어야 하는 것은 아니며 어떤 점이 되어도 무관하다. NFP의 형상은 두 다각형의 형상에 의해서만 결정되며 기준점의 위치는 NFP의 위치에만 영향을 주기 때문이다. 본 논문에서는 기준점이 그리는 자취가 실제의 작업 영역과 다른 2차원 영역에 그려지는 것이라 가정하고 이 영역을 기준점 맵이라 정의하였다. 다각형의 각 꼭지점들이 기준점에 대한 상대적인 좌표값을 가지고 있으면 기준점 맵 상에서의 한 점은 곧 실제 영역에서의 다각형의 한 위치를 정의할 수 있다.

Fig. 13는 기준점 맵이 사용되는 예를 보여주고 있다. Fig. 13(a)에서 A는 위치를 정하고자 하는 모델의 한 단면이고 B, C, D는 앞서 배치된 모델의 단면을 나타낸다. 이 검사 층에서 A가 B, C, D와 중첩되지 않으면서 놓일 수 있는 영역을 구해야 한다. 이 과정은 배치하고자 하는 삼차원 형상의 전체 단면에 대해 수행되며 그 결과는 계속 같은 기준점 맵 상에서 갱신된다. Fig. 13(b)에서 빗금친 부분은 A의 기준점이 작업 영역 내에서 놓일 수 있는 최대 영역 U를 나타낸다. 이는 A가 작업 영역의 내부에서 가장 자리에 접하면서 움직일 때 A의 기준점이 기준점 맵

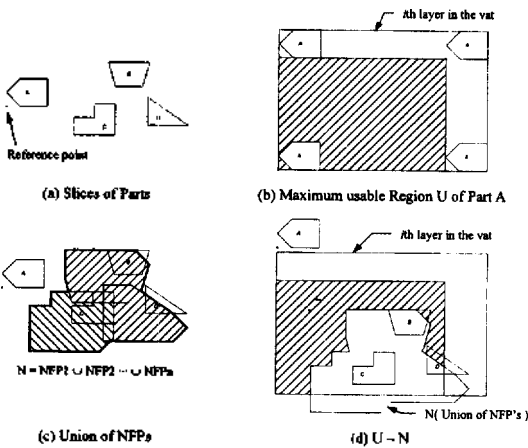


Fig. 13. Reference point map.

상에 그리는 자취와 같다. 작업 영역은 하나의 층을 나타내고 일반적으로 그 형상은 사각형이므로 최대 영역 U도 사각형의 형상을 갖게 된다.

기준점 맵 상에서 U를 구한 후 A가 B, C, D 각각의 다각형과 이루는 NFP를 기준점 맵 상에서 구한다. NFP는 실제 영역에서 A와 각 다각형간에 중첩이 생기는 영역에 대응이 되는데, 이들은 다각형의 합집합 연산을 통해 합집합 N에 더해진다. Fig. 13(c)에 NFP 및 N을 나타내었다.

NFP들의 합집합 N은 A의 기준점이 기준점 맵 상에서 놓일 수 없는 영역을 결정해 준다 따라서 Fig. 13(d)에서 기준점 맵 상의 빗금친 부분과 같은 $U-N$ ($U \setminus N$)의 영역은 A의 기준점이 놓일 수 있는 영역이 되며 이는 곧 실제 작업 영역에서 다른 다각형과 중첩이 발생하지 않는 영역에 대응이 된다. $U-N = \emptyset$ 인 경우는 A의 기준점이 놓일 수 있는 영역이 없는 것이며 이는 실제 작업 영역에서 A를 배치할 공간이 없는 것이다. 이러한 경우 A에 대해서는 다른 층에서부터 다시 중첩검사가 이루어져야 한다.

6.5 다각형 간의 합집합 연산

단면들간에 구해진 NFP들의 합집합을 구하기 위해 불리안 연산(boolean operation) 중의 하나인 합집합(union) 연산을 한다. 두 개의 다각형 A, B의 합집합을 구하는 기본적인 원리는 다음과 같다.

모서리는 데이터구조 내에 양 끝점의 좌표값을 가지며, 모서리의 끝점은 즉 자신을 시작점으로 하는 모서리를 데이터구조 내에 연결 리스트의 형태로 가도록 한다. 단면의 모서리는 Fig. 14(a)에서와 같이 외곽 루프는 반시계 방향, 내부의 루프는 시계방향으로 방향성을 가지도록 한다. 합집합을 구하고자 하는 두 단면의 정보를 읽어들이면 후에는 단면의 모든 모서리에 대해 교점을 구한다. 이때 다음의 세 가지 경우가 있을 수 있다.

- (1) 만나지 않는 경우
 - (2) 한 점에서 만나는 경우
 - (3) 교점의 개수가 무한히 많은 경우, 즉 모서리의 일부 혹은 전체가 중첩되어 있는 경우
- (2)와 같은 경우 모서리를 교점에서 나누고 교점은 모서리들이 공유하도록 한다. 교점이 모서리의 끝점과 일치할 경우는 모서리들이 그 점을 공유하도록 해 주기만 하면 된다. (3)의 경우는 무시하고 다음 모서리에서 교점 조사를 계속하면 된다. 이는 (3)의 경우에서 교점 공유가 되지 않더라도, 중첩하지 않는 다음 모서리간에 양 끝점에서 교점이 발생하여 공유가 일어나기 때문이다.

교점 조사가 끝나고 나면 Fig. 14(b)에서와 같이 하나의 다각형 위의 꼭지점 중 다른 다각형의 외부에 있는 점에서부터 출발해서 모서리의 방향을 따라가며 지표(flag)를 붙이도록 한다. 출발점을 결정하기 위해서는 점이 다각형의 내부에 있는 점인지 외부에 있는 점인지를 조사하는 내-외부 검사(in/out test)가 필수적이다. 출발점이 결정되고 나면 그 점에서 나가는 방향의 모서리를 따라가면서 지표를 붙인다.

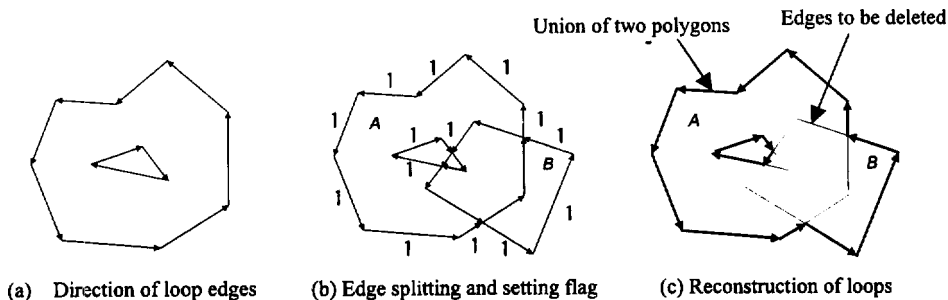


Fig. 14. Union of polygons.

꼭지점이 교점인 경우 그 점에서 나가는 방향의 모서리는 두 개 이상이 존재하게 된다. 하나의 점에서 두 개 이상의 모서리가 출발하고 있으면, 즉 꼭지점이 자신을 출발점으로 하는 모서리를 두 개 이상 가지고 있으면 모서리가 나가는 방향을 바라볼 때 가장 오른쪽으로 나가는 모서리를 따라가도록 한다. 이는 서로의 각도를 비교함으로써 찾을 수 있다. 이와 같은 과정을 모든 모서리에 대해 적용하고 난 후 같은 지표를 가지는 모서리들을 연결하면 Fig. 14(c)에 굵은 선으로 나타낸 것과 같이 두 다각형의 합집합에 해당하는 다각형의 최외곽 루프와 내부루프 등이 모두 구해진다.

6.6 3차원 공간 상에서의 모델 배치

2차원상에서의 중첩 검사는 앞 절에서 설명한 바와 같이 이루어지며, 단면 정보로 표현된 3차원 모델의 위치를 결정하기 위해서는 하나의 모델의 모든 단면들이 x, y 평면상에서 같은 기준점을 가지도록 해야 한다. 기준점들의 x, y 좌표가 같도록 설정되면 하나의 모델에 포함된 각 단면의 기준점은 같은 기준점 맵 상에 NFP를 그릴 수 있게 된다. 이를 위해, Fig. 15에 나타낸 것과 같이 바운딩 박스(bounding box)³에서 (X_{min}, Y_{min}) 인 점을 지나는 모서리를 기준축(reference axis)이라 하고 각 단면을 포함하는 평면과 기준축과의 교점을 각 단면의 기준점으로 설정

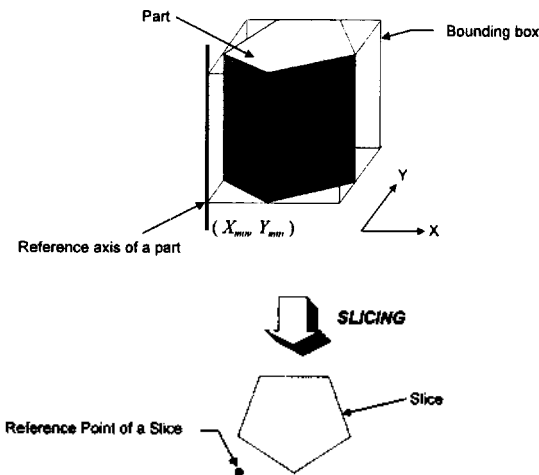


Fig. 15. Determination of reference point from bounding box.

³ 모델을 둘러싸는 직육면체 중 각 모서리가 x, y축에 평행하고 최소의 체적을 가지는 것을 바운딩 박스(bounding box)라 한다.

하였다.

각 단면이 그와 대응되는 층(corresponding layer)에 있는 단면들과 이루는 NFP들에 대해 합집합 연산을 수행해서 NFP의 합집합 N을 갱신한다. 이때 단면들의 기준점이 같은 x, y 좌표를 가지도록 설정되었으므로 서로 다른 층에서 구해진 NFP들이 하나의 영역 N에 더해질 수 있게 된다. N을 갱신할 때마다 전체 사용가능영역인 U와의 포함관계를 검사하도록 한다. 만약 Fig. 16에서와 같이 $U-N=\emptyset$ 이 되면, 즉 모델의 어느 한 단면이라도 대응층 내에 놓일 수 없는 것으로 판정이 되면 검사 대상층을 한 층 높인 후 다시 모델의 맨 아래 단면부터 중첩 검사를 시작한다.

모델의 모든 단면에 대해 NFP가 구해져서 N의 갱신이 끝날 때까지 $U-N \neq \emptyset$ 의 조건을 만족하거나, 검사 범위가 작업 공간 내에서 이미 사용된 층을 넘어 처음 사용되는 층으로 가게 되면 검사를 끝내고 모델을 배치할 수 있게 된다. 검사가 끝날 때까지 $U-N \neq \emptyset$ 의 조건을 만족하면 작업 공간에서 이미 사용된 층 내에 현재의 모델이 놓일 수 있는 공간이 있는 것임을 의미한다. 이때 U-N의 영역 중 Y_{min}, X_{min} 인 점, 즉 Y값이 최소인 점들 중 X값이 최소인 점에 기준축이 놓이도록 모델의 위치를 결정한 후 각 층별로 모델의 단면을 배치한다. U-N의 영역은 다각형이 되므로 꼭지점 중 가장 Y값이 작은 점을 찾고 그러한 점이 두 개 이상인 경우 그 점들 중 X값이 가장 작은 점을 찾으면 된다. 검사 범위가 작업 공간 내에서 사용된 층을 넘어가는 것은 현재의 모델이 새로운 층을 사용하게 됨을 의미한다. 이 때에도 검사가 이루어진 층까지의 N에서 기준축이 놓일 위치(Y_{min}, X_{min} 인 점)를 찾아 모델을 배치하도록 한다.

이상과 같은 알고리즘을 통해 2차원 작업 영역 내에 다각형을 배치하는 과정을 Fig. 17에 나타내었다. Fig. 17(a)에서 다각형 E는 BL 조건에 따라 배치하고자 하는 다각형이고, 나머지 다각형들은 이미 배치가

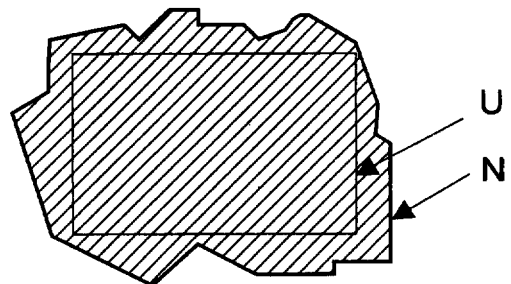


Fig. 16. Example of the case, $U \subset N$ ($U \neq N$).

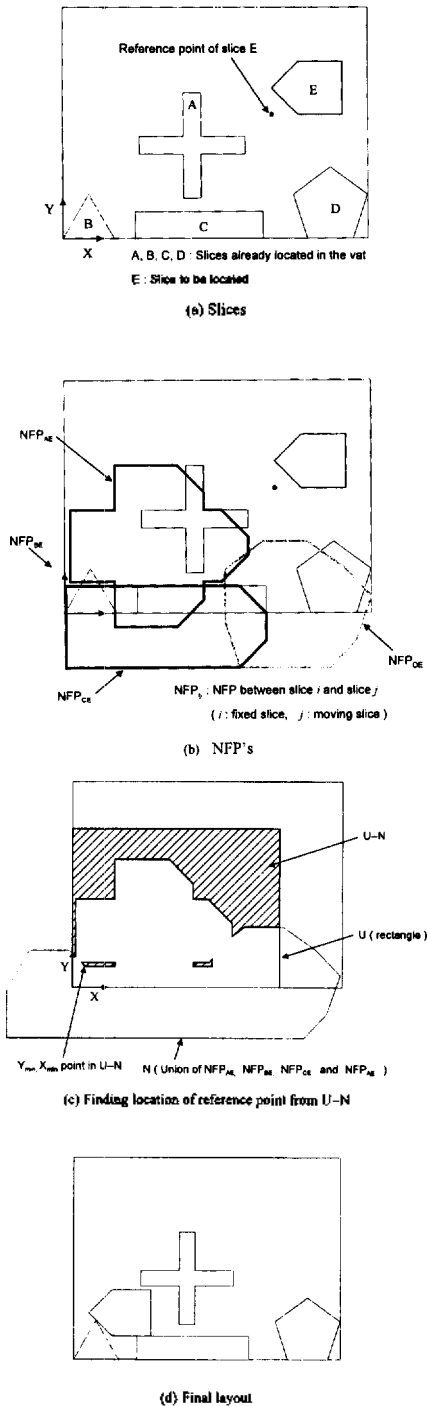


Fig. 17. Example of determining the location of a slice in 2D space.

된 다각형이다. 여기에서의 2차원 작업 영역은 임의의 층을 고려한 것이므로 이미 배치된 단면 A, B, C,

D는 그림에서와 같이 겹치지 않을 수 있다. Fig. 17(b)에서와 같이 각 다각형과 E가 이루는 NFP를 구하고 나면 기준점 맵 상에서 Fig. 17(c)와 같은 U-N의 영역을 구할 수 있다. 이 영역에서 Y_{min} , X_{min} 의 위치를 찾고 그 점에 E의 기준점을 두게 되면 Fig. 17(d)와 같이 BL조건을 만족하는 위치에 E가 배치된다.

3차원 모델의 경우에는 모델의 모든 단면에 대해 같은 X, Y값을 갖도록 기준점을 설정하고 모든 단면에 대해 하나의 기준점 맵을 구성하여 위치를 결정된 후 좌표 변환 벡터를 구하면 된다. 이러한 과정을 Fig. 18에 나타내었다. 2차원에서의 패킹 알고리즘이 3차원으로 확장된 과정을 간략히 나타낼 수 있도록 모델은 3개의 단면을 갖는 것으로 가정하였고 단면의 형상도 임의로 설정하였다. Fig. 18(a)에서 회색의 다각형은 배치하고자 하는 모델의 단면이고 나머지 다각형들은 이미 배치가 된 모델들의 단면이다. 배치가 된 모델의 단면은 한 개 이상의 층에서 다른 모델의 단면에 접해 있어야 한다. Fig. 18(b)는 기준점 맵이 갱신되는 과정을 나타내고 있으며 각 층에 대해 NFP가 모두 구해진 후의 U와 N이 그려져 있다. 빗금이 쳐진 부분은 NFP의 합집합 N을 나타낸다. Fig. 18(c)는 모든 단면에 대해 NFP가 구해지고 $U-N \neq \emptyset$ 이 되어 중첩 검사가 끝난 후의 기준점 맵이며 U-N 영역에서 모델의 위치를 결정된 후 각 층에 모델의 단면을 배치한 그림이 Fig. 18(d)에 나타나 있다.

이상과 같은 과정을 통해 작업 공간내의 위치를 결정된 후에는 모델을 배치해야 한다. 위치가 결정된 모델을 배치하는 것은 곧 모델의 각 단면을 작업 공간 내의 해당 층에 배치하는 것과 같다. 이를 위해 우선 좌표 변환 벡터를 구한 후, 모델의 각 단면을 이루는 다각형의 각 꼭지점의 좌표를 구하고 이를 자료 구조에 저장하는 과정을 수행한다. 작업 공간을 나타내는 자료 구조는 모델의 정보를 저장하는 방식과 유사하며 각 층을 나타내는 연결 리스트에는 그 층에 배치된 단면들의 정보가 저장된다. 단면의 정보에는 작업 공간의 좌표계에 대한 각 꼭지점의 좌표값이 연결 리스트로 저장되어 있고 자신이 속한 모델의 고유값이 저장되어 있다.

7. 최적화 기법의 적용

방향이 결정된 모델의 위치는 이상과 같은 알고리즘으로 결정하도록 하고, 전체 사용 층의 높이가 최소가 되도록 하는 모델의 배치순서, 모델의 방향과

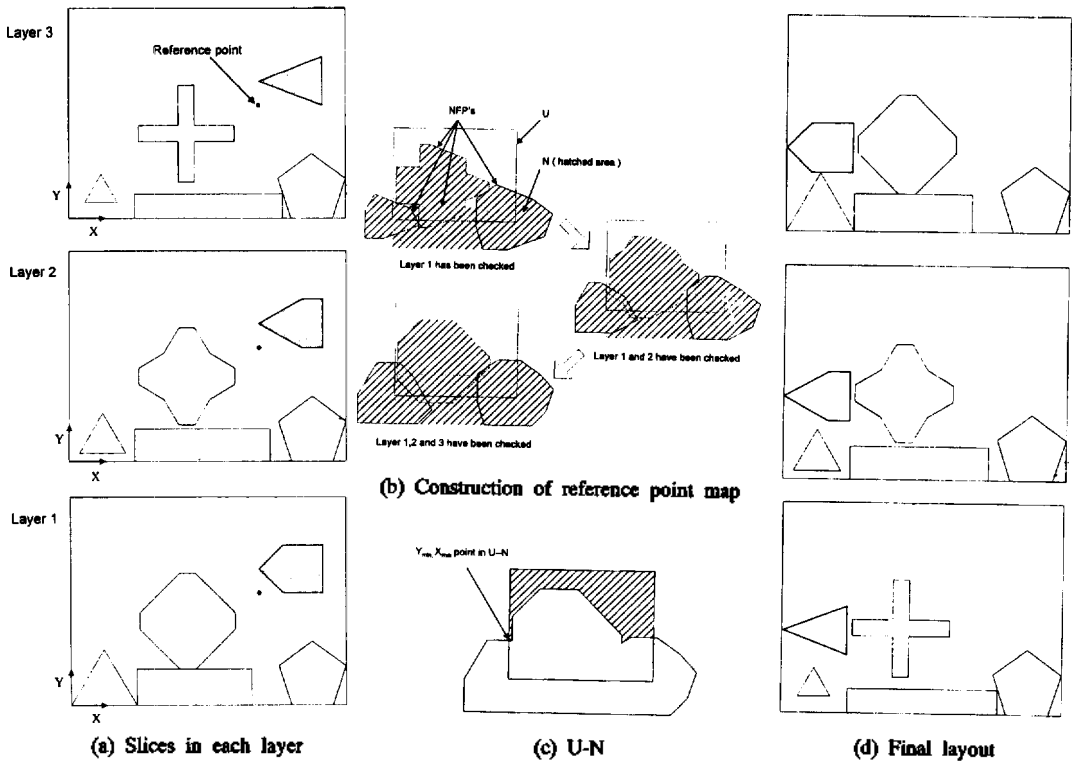


Fig. 18. Simple example of three-dimensional packing process.

같은 변수는 최적화 기법을 이용하여 결정하도록 하였다. 최적화 기법으로는 유전 알고리즘(genetic algorithm)을 적용하였다.

7.1 유전 알고리즘

유전 알고리즘은 자연 진화의 법칙인 적자 생존과 자연 도태의 원리를 토대로 하여 정립한 최적화 알고리즘이다. 자연 진화의 법칙에 의하면 일정한 환경에 무리를 지어 살고 있는 생명체들은 그 환경에 적합한 형질을 가진 개체가 생존할 확률이 크며, 교배와 변종의 과정을 거쳐 더 좋은 방향으로 진화해 나가고, 부적합한 형질의 개체들은 진화의 과정에서 점차 도태되어 간다. 이와 같은 진화의 과정을 되풀이하면 결국 주어진 환경에 가장 적합한 형질의 개체들이 형성될 것이다.

유전 알고리즘은 이러한 자연 진화 법칙을 일반 최적화 문제에 적용한 방법으로, 해석 영역에 다수의 객체들을 분포시켜 목적 함수값과 제한 조건의 위반 정도에 따라 각 객체에 적합성을 부여한다. 객체의 적합성이 클 수록 다음 단계인 교배와 변종의 과정에 참여할 확률을 다른 객체보다 크도록 하면

적합성이 좋은 객체들이 다음 단계에 보다 많이 형성되어, 계산이 진행될 수록 전체 객체들은 재생성(reproduction), 교배(crossover), 변종(mutation) 등의 과정을 통해 적합성이 좋은 방향으로 탐색이 진행되게 된다.

7.2 유전자 구조의 설정

본 알고리즘의 구현에서 필요한 최적화 과정 변수로는 모델의 배치순서, 각 모델의 방향 결정을 위한 변수가 있다. 모델의 방향을 결정하기 위해서는 Z축에 대한 회전 각도가 주어져야 하고 모델의 위아래를 바꿀 것인지의 여부를 나타내는 값도 주어져야 한다. 유전자는 이러한 변수값들을 코드 내에 가지고 있도록 해야 한다.

유전 알고리즘의 변수형으로는 정수형 변수를 사용하도록 하며 유전자는 다음의 Fig. 19에서와 같은 구조를 갖도록 하였다.

유전자의 앞부분에는 모델을 배치하는 순서를 나타내는 순열이 저장된다. 각 모델의 방향을 나타내는 정수값은 유전자의 뒷부분에 저장되며 z축에 대한 회전 각도와 상하 반전 여부를 함께 나타낼 수 있

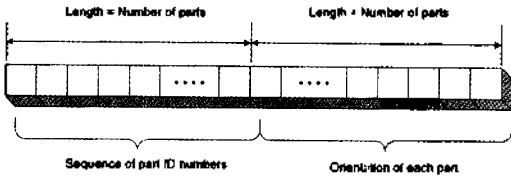


Fig. 19. Structure of genetic string.

Table 1. Orientation variables

Orientation variable	1	2	3	4	5	6	7	8
Rotation angle (degree)	0	90	180	270	0	90	180	270
Upside down	No	No	No	No	Yes	Yes	Yes	Yes

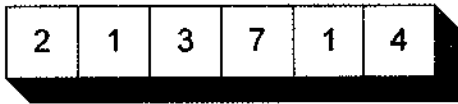


Fig. 20. Example of genetic string.

도록 한다. 각 모델에 대한 방향 변수값의 범위는 사용자가 설정한 최소 회전 각도에 따라 달라진다.

예를 들어 3개의 모델 P₁, P₂, P₃가 있고 최소 회전 각도는 90도로 설정하였다고 하면 유전자 앞부분의 세 코드는 1에서 3까지의 순열을 가지며 뒷부분의 세 코드는 각각 1에서 8까지의 방향 변수값을 가질 수 있다. 이 때 방향 변수값이 나타내는 의미는 다음의 Table 1과 같다.

예를 든 3개의 모델에 대한 패키징 최적화 과정 중 다음의 Fig. 20과 같은 유전자가 얻어졌다고 하자. 이는 3개의 모델을 배치하는 순서가 P₂, P₁, P₃이며 P₁, P₂, P₃의 각 모델의 방향 변수값이 7, 1, 4임을 나타낸다. 각 방향 변수값에 의한 모델의 방향 설정은 Table 1에 나타낸 것과 같게 된다.

7.3 목적함수의 정의

전체 최적화 과정을 지배하는 목적함수(objective function)에는 최적화하고자 하는 변수가 포함되어야 한다. 본 논문에서 다루고 있는 문제에서 가장 중요한 변수는 패키징이 끝난 후에 모델들에 의해 사용된 높이이다. 이 값은 사용된 층의 개수를 사용하기로 한다.

본 알고리즘에서 사용된 목적 함수는 식 (1)과 같은 형태를 가지도록 하였다. 최적화 과정은 이 목적 함수의 값을 최소화하는 방향으로 진행되었다.

$$f_{obj} = w_1 \times f_1(l_h) + w_2 \times f_2(l_x, l_y) \quad (1)$$

w₁, w₂: weight values

l_h: height of the packed volume

l_x, l_y: x,y length of the bounding box enclosing all the parts

우변에서 첫번째 항은 높이의 함수로서 높이 l_h값이 작을수록 f₁(l_h)의 값도 작도록 설정되어야 하며, 두번째 항은 전체 모델을 둘러싸는 바운딩 박스(bounding box)를 x-y평면상에 투영한 사각형의 면적에 대한 함수로서 역시 그 면적이 작을수록 함수값도 작은 값을 갖도록 해야 한다. 특히 두번째 항은, 높이 l_h가 같은 경우에 패키징효율을 비교할 수 있는 지표가 된다. 높이가 낮도록 패키징하는 것이 가장 중요한 목적이 되므로 가중치는 w₁이 w₂보다 큰 값을 가지도록 한다. 본 논문에서는 f₁(l_h)과 f₂(l_x, l_y)를 다음과 같이 정의하였다.

$$f_1(l_h) = s \times \frac{l_h}{H} \quad (2)$$

s: scale factor

H: height of the vat

$$f_2(l_x, l_y) = \frac{l_x \times l_y}{W \times D} \quad (3)$$

W: width of the vat

D: depth of the vat

적합성(fitness)은 목적함수를 통해 계산된 값을 기초로 계산이 되며 유전자의 적합성이 클 수록 그 유전자는 좋은 것으로 판정이 된다.

8. 적용 예

Fig. 21에 나타낸 16개의 모델을 각각 2개씩 사용한 32개의 모델에 대해 본 논문에서 제시한 패키징 알고리즘을 적용하고 최적화를 수행하였다. 객체지향 언어인 C++를 이용하여 구현하였으며 UNIX기반의 SGI의 Indigo2 XZ에서 작업이 수행되었다. 작업 공간의 크기는 350×500×500 mm³이며, 각 모델은 Z축에 대해 90°씩 회전할 수 있게 하였다. 계산량을 줄이기 위해 패키징 하기 전에 각 모델의 단면 정보를 간략화하였다. 즉 여러 층을 하나의 층으로 합함으로써 패키징 알고리즘에 주어지는 데이터는 실제 단면 정보보다 층 두께가 큰 값을 가지도록 하는 것이다. 여러 층을 합하는 것은 앞서 서술한 다각형의 합집합 연산을 적용하면 된다.

층 간에 형상 변화가 큰 경우에는 패키징 후에 실제 모델을 배치하면 모델 간의 약간의 공간이 발생할 수 있지만, 가공 후 모델을 분리하기 위해 모델간에

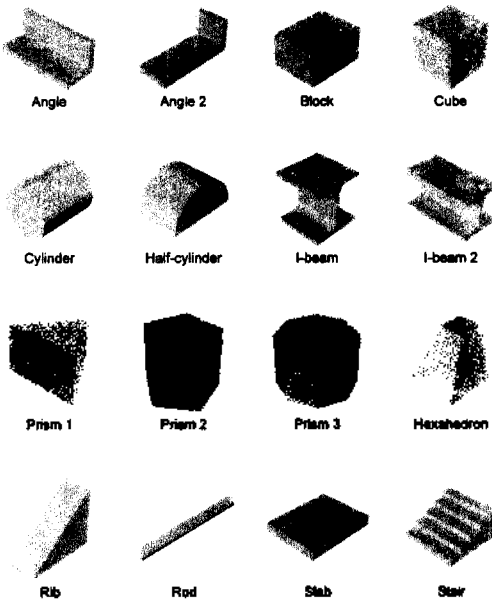


Fig. 21. Example models.

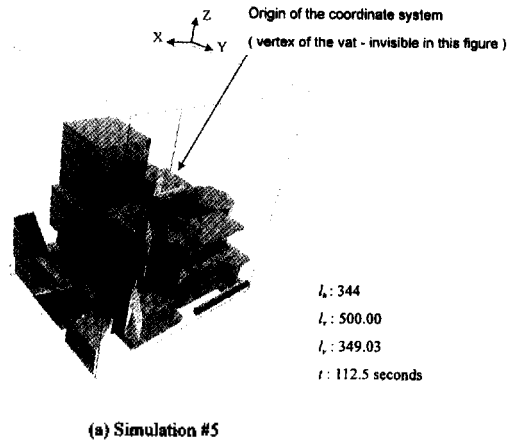
어느 정도의 간격이 있어야 함을 고려하면 그러한 공간은 무시할 수 있을 것이다. 간략화를 할 때 합하게 될 층의 개수는 모델들을 분리할 때 필요한 간격을 고려하여 설정한다.

Fig. 22에 렌더링된 STL 이미지로 그 결과를 순차적으로 나타내었으며 모델이 차지하는 높이가 점차 낮아지고 xy 평면에 바운딩 박스(bounding box)를 투영한 넓이도 작아지는 것을 볼 수 있다. 각 모델은 알아보기 쉽게 하기 위하여 투명도를 주고 렌더링을 수행하였다. Fig. 22에 나타난 'simulation #'는 최적화 수행을 위하여 생성된 총 유전자들의 개수를 의미한다.

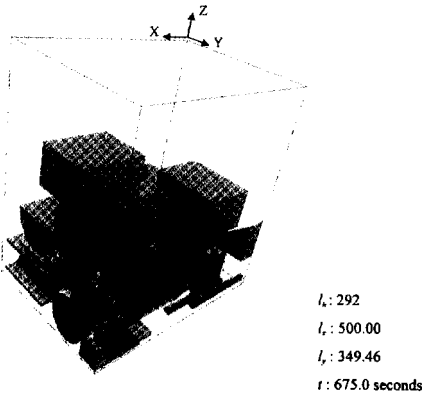
좀 더 실제적인 모델들에 대하여 본 알고리즘을 적용해 보았다. 본 예제들은 Cubital[®]의 Solider5600에서 성형하기 위하여 준비된 모델들이며 최적화를 수행하기 위하여 주어지는 사용자 입력값들과 제한조건들은 아래와 같다.

System input:

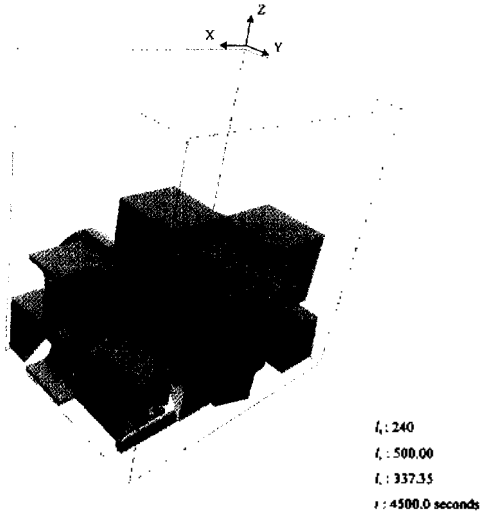
- Vat size: 350×500×500 mm³
(for Cubital[®] machine)
- Slicing thickness(Input file): 5 mm
- Slice files of multiple parts: 32
- Minimum rotational angle: 90°
- Weight for height influence, w_1 : 1.0
- Weight for area influence, w_2 : 0.7
- Scale factor, s: 4.0
- Number of iterations: 1000



(a) Simulation #5

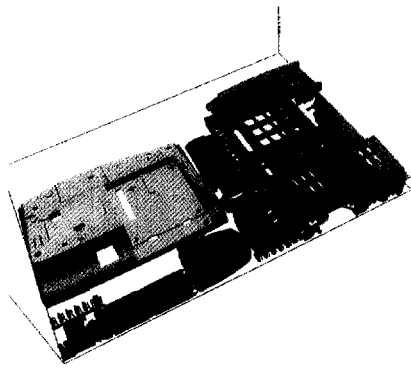


(b) Simulation #30

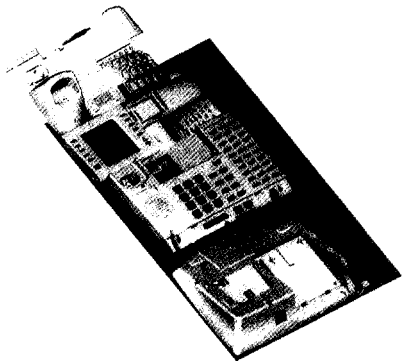


(c) Simulation #200

Fig. 22. Results of the three-dimensional packing.



(a) Our result



(b) Magics® RP result

Fig. 23. Results of nesting practical multiple parts.

Fig. 23에는 본 알고리즘에 의해서 수행된 최적화의 결과가 상용 소프트웨어인 Magics[®] RP를 이용하여 같은 모델을 배치시켰을 때의 결과와 비교하여 도시되어 있다. 그림에서 보는 바와같이 Magics[®] RP의 자동 배치기능은 기본적으로 2차원 패키징만을 지원하기 때문에 RP 기계의 플랫폼의 면적이 주어진 파트를 모두 배치할 만큼 충분히 크지 않을 경우에는 플랫폼 바깥쪽으로 물체의 일부가 배치되게 된다.

Fig. 23에 쓰여진 모델들을 Cubital[®]사의 Solid Ground Curing 시스템에서 한 번의 공정으로 성형하기 위하여 시스템과 함께 제공되는 소프트웨어 환경에서 전문가가 완전 수동으로 3차원 배치를 할 경우에 대략 5시간 정도가 소요된 반면에 본 알고리즘에 의하여 패키징을 수행했을 경우에는 Fig. 24와 같이 최적화 과정이 완전히 수렴할 때까지 약 4350초(1시간 12분 30초) 정도가 소요되었다.

9. 결 론

본 연구에서는 RP 가공에서 모델의 단면 데이터

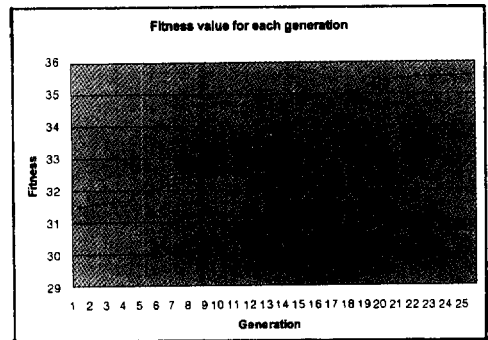


Fig. 24. Convergence status of optimization process for example parts.

를 이용해서 3차원 패키징을 수행한 새로운 알고리즘을 구현하였다. 3차원 모델의 단면 정보를 이용해 2차원상에서 수행된 본 알고리즘은 모델의 형상이 불록하거나 오목함에 관계없이 모델을 중첩 없이 배치할 수 있다. NFP를 2차원 단면의 패키징에 적용함으로써 중첩이 없이 모델이 놓일 수 있는 영역을 효과적으로 찾을 수 있었다.

본 알고리즘은 RP 가공에서의 패키징 문제에 대해 RP 가공이 모델의 단면 정보를 이용한다는 특성에 착안하여 2차원에서의 패키징 알고리즘을 그대로 활용한 특징을 가지고 있다. RP가공은 가공 방법 및 재료등에 따라 여러 종류가 있지만 모델의 단면 정보를 이용하여 3차원 모델을 성형하는 것은 공통된 특성이므로 본 알고리즘은 일반적인 RP 가공에 적용할 수 있다. 단, SLA나 FDM과 같이 성형 시 지지대가 필요한 가공에 대해서는 지지대가 다른 물체의 표면에 생성되는 경우를 배제할 수 있도록 구속 조건을 적용해야 할 것이다. 즉 다른 모델의 위에 모델의 일부분 혹은 전체가 놓이지 않도록 하는 것이다. 이는 모델의 형상 정밀도에 영향을 주지 않기 위함이다. 만약 그러한 배치가 허용된다면, 패키징이 완료된 후에 지지대를 생성하도록 해야 할 것이다.

본 알고리즘이 성공적으로 수행이 되는 것으로 확인되었지만, 전체적으로 소요되는 시간을 줄일 수 있도록 알고리즘의 보강이 요구된다. 빠르고 안정된 다각형 불리안 연산이 필요하며 데이터를 단순화하는 기법도 적용되어야 할 것이다. 또한 유전 알고리즘을 이용한 최적화 과정에서는, 본 연구에서 제시한 패키징 알고리즘에 대해 효율적인 결과를 가져올 수 있는 유전 연산자의 개발이 필요하다.

후 기

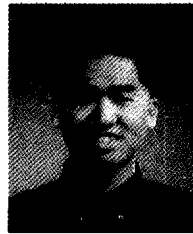
본 연구는 서울대학교부설 정밀기계설계공동연구소와 터보동력기계연구센터의 지원 하에 이루어진 것입니다.

참고문헌

1. Jacobs, P.F., *Rapid Prototyping & Manufacturing: Fundamentals of StereoLithography*, Society of Manufacturing Engineers, 1st Ed., 1992.
2. Adamowicz, M. and Albano, A., "Nesting Two-Dimensional Shapes in Rectangular Modules", *Computer Aided Design*, pp. 27-33, 1976.
3. Ismail, H.S. and Hon, K.K., "New Approaches for the Nesting of Two-Dimensional Shapes for Press Tool Design", *Int. J. Prod. Res.*, Vol. 30, No. 4, pp. 825-837, 1992.
4. Cho, K.H. and Lee, K.W., "New Nesting Approach for the Shape Allocation on Several Irregular Raw Sheets", Ph.D dissertation, 1994.
5. Ismail, H.S. and Hon, K.K., "The Nesting of Two-Dimensional Shapes Using Genetic Algorithms", *Proc. Instn. Mech. Engrs.* Vol. 209 Part B: Journal of Engineering manufacture, pp. 115-124, 1995.
6. Jakobs, S., "On Genetic Algorithms for the Packing of Polygons", *European Journal of Operational Research*, pp. 165-181, 1996.
7. Jain S. and Gea, H.C., "Two-Dimensional Packing Problems Using Genetic Algorithms", *Proceedings of The 1996 ASME Design Engineering Technical Conference and Computers in Engineering Conference*, pp. 1-8, 1996.
8. Albano A. and Sapuppo, G., "Optimal Allocation of Two-Dimensional Irregular Shapes Using Heuristic Search Method", *IEEE Transactions on Systems, Man, And Cybernetics*, Vol. SMC-10, No. 5, pp. 242-248, 1980.
9. Wodziak, J.R., Fadel, G.M. and Kirschman, C., "A Genetic Algorithm for Optimizing Multiple Part Placement to Reduce Build Time", *The Fifth International Conference on Rapid Prototyping*, pp. 201-210, 1994.
10. *Guide to Magics® RP V4.1*, Materialise N.V., 1997.
11. 허정훈, 이건우, "SLA를 이용한 신속시작 작업에서 최적성형방향의 결정", 한국정밀공학회지 제13권

제4호, pp. 163-172, 1996.

12. *Proceedings of SIGGRAPH 93*, Anaheim, California, Aug 1-6, pp. 377-378, 1993.
13. Ashley, S., "Rapid Prototyping Systems", *Mechanical Engineering*, Vol. 113, No. 4, pp. 34-43, 1991.
14. Thomas, S.W., "StereoLithography Simplifies Tooling for Reinforced Rubber Part", *Mechanical Engineering*, pp. 62-66, July, 1992.
15. Hur, J.H. and Lee, K.W., "Development of CAD Environment to Determine Preferred Build-up Direction for Layered Manufacturing", Accepted by *Journal of Advanced Manufacturing Technology*, 1997.



허 정 훈

1994년 서울대학교 기계설계학과 학사
 1996년 서울대학교 기계설계학과 석사
 1996년 ~ 현재 서울대학교 기계설계학과 박사과정
 관심분야 : Rapid prototyping, Solid modeling, Reverse engineering, Computer graphics



이 건 우

1978년 서울대학교 기계공학과 학사
 1981년 미국 MIT 공학석사
 1984년 미국 MIT 공학박사
 1984년 ~ 1986년 Univ. of Illinois at Urbana-Champaign 조교수
 1986년 ~ 현재 서울대학교 기계설계학과 교수
 관심분야 : Computer aided geometric design, NC tool path generation and verification, Solid modeling, Rapid prototyping



안 재 홍

1994년 서울대학교 기계설계학과 학사
 1997년 서울대학교 기계설계학과 석사
 1997년 ~ 현재 한국과학기술연구원 CAD/CAM 연구팀 연구원
 관심분야 : Rapid prototyping, Computer graphics, Virtual reality