

로봇 팔레타이징 시뮬레이터를 위한 적재 패턴 생성 및 시변 장애물 회피 알고리즘의 제안

Algorithmic Proposal of Optimal Loading Pattern and Obstacle-Avoidance Trajectory Generation for Robot Palletizing Simulator

유 승 남, 임 성 진, 한 창 수*, 김 성 락
(Seung-Nam Yu, Sung-Jin Lim, Chang-Soo Han, and Sung-Rak Kim)

Abstract : Palletizing tasks are necessary to promote efficient storage and shipping of boxed products. These tasks, however, involve some of the most monotonous and physically demanding labor in the factory. Thus, many types of robot palletizing systems have been developed, although many robot motion commands still depend on the teach pendant. That is, the operator inputs the motion command lines one by one. This is very troublesome and, most importantly, the user must know how to type the code. We propose a new GUI(Graphic User Interface) for the palletizing system that is more convenient. To do this, we used the PLP “Fast Algorithm” and 3-D auto-patterning visualization. The 3-D patterning process includes the following steps. First, an operator can identify the results of the task and edit them. Second, the operator passes the position values of objects to a robot simulator. Using those positions, a palletizing operation can be simulated. We chose a widely used industrial model and analyzed the kinematics and dynamics to create a robot simulator. In this paper we propose a 3-D patterning algorithm, 3-D robot-palletizing simulator, and modified trajectory generation algorithm, an “overlapped method” to reduce the computing load.

Keywords : palletizing robot, robot trajectory optimization, robot palletizer, overlap method

I. 서론

산업현장에서 팔레타이징 작업 공정은 일반적으로 팔레타이징 작업 전용 로봇이나 6축 다관절 로봇을 사용하여 구축된다. 다른 자동화 공정과 마찬가지로 로봇을 사용하는 이유는 뛰어난 유연성 때문이라고 할 수 있다[1]. 이러한 유연성에 의한 잠재력은 산업용 로봇이 쉽게 프로그램이 되는 경우에만 충분히 실현될 수 있다. 간단 작업을 위해서 불필요한 프로그래밍 코드가 늘어나거나 사용자 편의성 또는 자유도가 결여된 로봇 프로그램은 해당 공정의 비용과 직결되는 문제를 야기할 수 있다. 본 연구의 최종 목표는 사용자 지향적 로봇 팔레타이징 오프라인 소프트웨어(robot palletizing off-line S/W)를 개발하는 것으로서, 이를 위해서는 기본적으로 팔레타이징 자동화 시스템은 로봇을 이용하며 단순한 로봇의 제어만이 아닌 대상물이 이송되는 컨베이어 시스템이나 팔레트(pallet), 간지(slip sheet)와 같은 팔레타이징 구성물들을 동시에 고려한 로봇 구동 알고리즘이 요구된다. 본 논문에서는 이와 같은 팔레타이징 적재 시스템의 특징을 고려한 적재 자동화용 OLP(Off Line Programming) S/W를 제안한다. 그림 1은 이러한 S/W의 기본 구조를 도시하고 있다.

본 논문에서는 그림 1에 도시한 바와 같이 ‘fast algorithm’ [2]을 이용하여 사용자가 입력한 적재물 및 팔레트의 크기에 최적화된 패턴을 실시간으로 생성하고 그에 따른 로봇의 경로 최적화를 고려한 팔레타이징 적재 경로의 생성을 통해 적재 작업을 시뮬레이션 할 수 있는 로봇 팔레타이징 적재 시

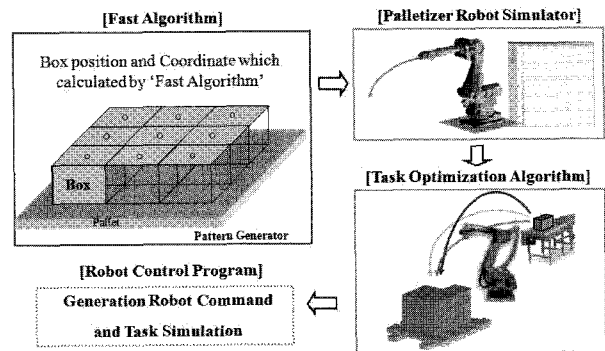


그림 1. 제안된 시뮬레이터의 기본 구조.
Fig. 1. Basic concept of proposed simulator.

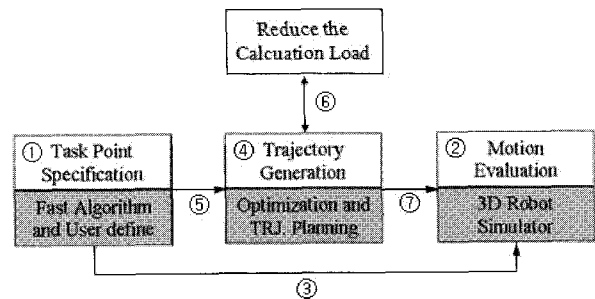


그림 2. S/W 개발 순서.
Fig. 2. S/W development process.

뮬레이션 프로그램을 제안한다.

그림 2는 제안된 S/W의 개발 프로세스를 나타내고 있다. 먼저, 사용자가 작업물의 크기 및 작업 컴포넌트들(로봇, 팔레트, 컨베이어벨트 등)의 위치를 입력하면 패턴생성시뮬레이터(그림 2 ①)는 적재물의 적재패턴을 생성하고, 로봇이 작업

* 책임저자(Corresponding Author)

논문접수 : 2007. 4. 13., 채택확정 : 2007. 9. 11.

유승남, 임성진 : 한양대학교 기계공학과 대학원

(hymecer@gmail.com/brass_79@hotmail.com)

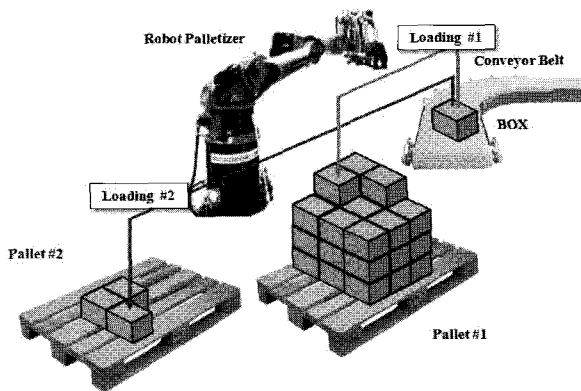
한창수 : 한양대학교 기계공학과 교수(cshan@hanyang.ac.kr)

김성락 : 현대중공업 로봇개발실 실장(ksr@hhi.co.kr)

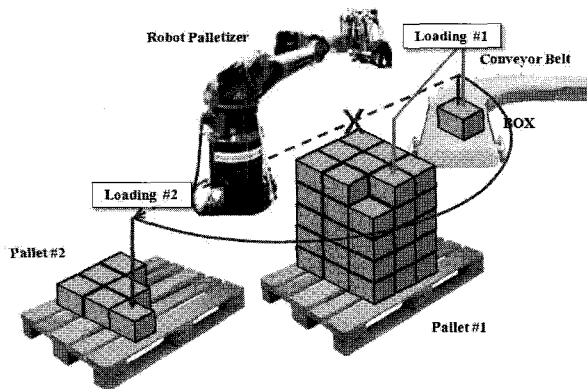
할 좌표 값을 생성한다. 다음 단계로 시뮬레이터 상에 로봇 모델을 생성(그림 2 ②)하고, 위에서 생성한 각 컴포넌트들의 좌표 값을 넘겨받는다(그림 2 ③). 시뮬레이터는 위의 경로 좌표 값을 기준으로 로봇시뮬레이터 내부에서 로봇의 작업 경로를 생성(그림 2 ④)하고, 로봇의 궤적상에 모든 작업 컴포넌트를 반영하여 로봇의 궤적을 생성 한다(그림 2 ⑤).

여기에, 본 연구에서는 추가로 본 논문에서 제안한 overlap method를 사용하여 연산시간을 최소화한 개량된 로봇궤적을 생성(그림 2 ⑥)하고 이를 시뮬레이션으로 가시화한다(그림 2 ⑦).

로봇 팔레타이징 작업을 위한 기본 공간은 ‘셀(cell)’로서 정의되며 이러한 셀은 몇 가지 기본 장비로 구성된다. 기본 구성물은 일반적으로 작업물을 셀에 공급하는 컨베이어 벨트, 작업물을 적재하기 위한 팔레트, 적재된 작업물이 무너지는 것을 방지하기 위한 간지, 적재 작업을 수행하기 위한 로봇 등의 4가지로 구성된다. 여기서 주목할 점은 이러한 기본 작업 구성물들은 팔레타이징 작업을 위한 기본 구성물인 동시에, 로봇의 측면에서는 장애물이 될 수 있다는 것이다. 일례로, 여러 개의 팔레트에 동시에 적재작업을 수행하는 경우, 그림 3과 같이 전치되어 있는 팔레트 및 적재물은 시간에 따라



(a) 병렬 적재의 작업 초기
(Loading #1, Loading#2: 정상 궤적으로 적재 가능)



(b) 병렬 적재의 작업 경과 후
(Loading #1: 정상궤적, Loading#2: 회피궤적)

그림 3. 로봇 팔레타이징 작업 시 팔레트 적재물이 장애물로 작용하는 사례.

Fig. 3. Example of loaded subjects functioning as obstacles to the robot-palletizing task.

형상이 변하는 이른바, ‘시변 장애물’이 된다. 따라서 팔레타이징 적재를 위해 필요한 주변 구성요소는 로봇에게는 언제 든지 장애물이 될 수 있는 것이다.

II. Fast algorithm 을 이용한 패턴 생성 시뮬레이터 개발

1. Steudel's heuristic algorithm

본 연구에서 사용한 fast algorithm은 기본적으로 Steudel이 제안한 기존의 4-block pattern heuristic algorithm을 개선한 것이다. Steudel's heuristic algorithm의 기본 패턴은 그림 4와 같다. 여기서 W 는 팔레트의 너비, L 은 팔레트의 길이를 의미하고, W_i 와 L_i 는 각각 팔레트를 영역별로 블록화 하였을 때의 i 번째 블록의 너비와 길이를 나타낸다.

Steudel's algorithm은 각각의 블록에 박스를 적재할 때 같은 방향으로만 적재하는 방법으로서 1, 3번 블록은 가로방향으로 박스를 적재하고 2, 4 블록은 세로방향으로 박스를 적재한다. Steudel's algorithm은 2단계로 구성되어 있으며, 첫 번째 단계에서는 그림 5와 같이 4가지 패턴의 초기 해를 생성한다.

이 과정에서 P2, P4와 같이 불가능한 패턴이 발생하게 되는데, 두 번째 단계에서 초기 해를 개선하는 과정을 거치게 된다. 즉, 그림 6에 도시한 바와 같이 (treatment1) 3,4번 블록을 고정시키고 1,2번 블록의 길이를 변경하여 해를 개선하는 방법과, (treatment2) 1,2번 블록을 고정시키고 3,4번 블록의 길이를 변경하여 해를 개선하는 방법을 비교하여 더 좋은 해를 선택한다.

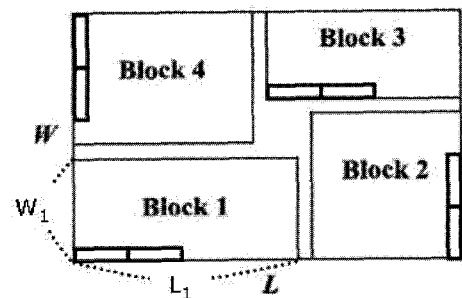


그림 4. Steudel 알고리즘.

Fig. 4. Steudel's heuristic algorithm.

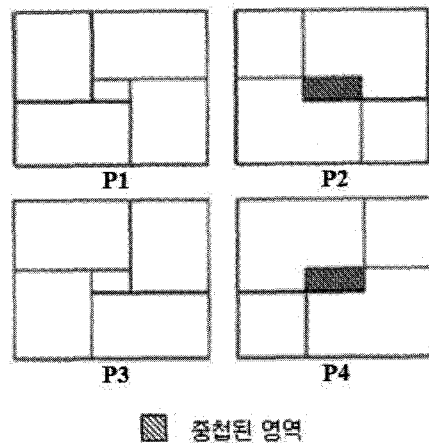


그림 5. Steudel's algorithm의 초기해 생성 방식.

Fig. 5. Initial treatment generation using Steudel's algorithm.

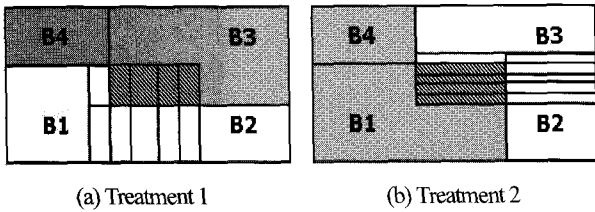


그림 6. Stuedel's algorithm의 해의 생성.
Fig. 6. Treatment of Stuedel's algorithm.

2. The fast algorithm

본 연구에서 사용한 fast algorithm은 Stuedel's algorithm과 같이 4가지 패턴의 초기 해를 생성하되, Stuedel's 처럼 블록의 모든 크기를 고려하지 않아 algorithm수행 속도를 빠르게 하였다(그림 7). 여기서 l 은 적재물의 길이, w 는 적재물의 너비를 나타낸다.

Fast algorithm의 첫 번째 단계에서는 1번 블록의 크기를 $(\bar{a}, \bar{b}), (\underline{a}, \underline{b}), (\underline{a}, \bar{b})$ 와 (\bar{a}, \underline{b}) 의 4개의 조합만을 고려하여 계산한다. 두 번째 단계에서는 초기 해를 개선하게 되는데, Stuedel's algorithm의 treatment 1과 2에 새롭게 treatment 3 추가하여 중공(central hall)을 만들어 중첩영역을 제거하는 과정을 거친다(그림 8).

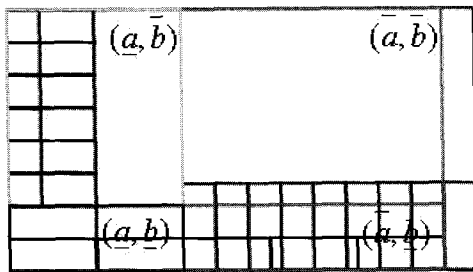


그림 7. Fast algorithm의 파라미터.
Fig. 7. Parameters of the fast algorithm.

- \bar{a} : block의 길이를 최대화하면서 box를 길이 방향으로 배열했을 때 최대로 적재 가능한 box수 = $5l$
- \underline{a} : block의 길이를 최소화하면서 box를 길이 방향으로 배열했을 때 최소로 적재 가능한 box수 = $2l$
- \bar{b} : block의 너비를 최대화하면서 box를 너비 방향으로 배열했을 때 최대로 적재 가능한 box수 = $8w$
- \underline{b} : block의 너비를 최소화하면서 box를 너비 방향으로 배열했을 때 최소로 적재 가능한 box수 = $2w$

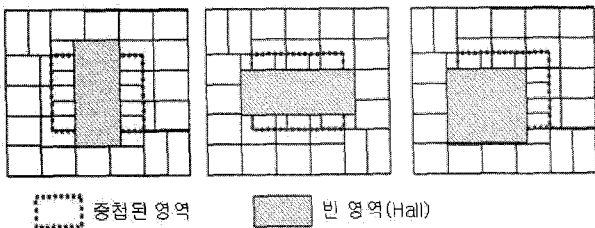


그림 8. The fast algorithm의 3단계 treatment.
Fig. 8. Three-step treatment of the fast algorithm.

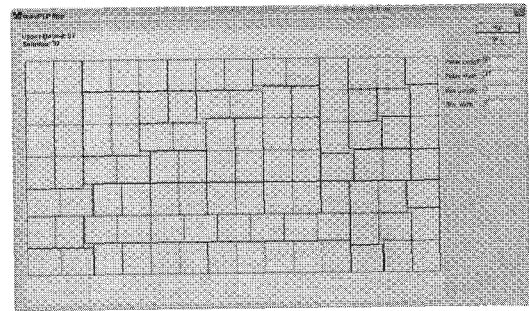


그림 9. Fast algorithm을 이용한 패턴 생성 프로그램.
Fig. 9. Pattern generation program using the fast algorithm.

표 1. 적재물 패턴 생성 시뮬레이션 결과.

Table 1. The test results of small sizes.

(L, W, l, w)	Loaded boxes	Elapsed Seconds
(700,234,19,8)	1077	0.2
(700,234,19,7)	1231	0.6
(700,234,18,9)	1001	0.1
(700,234,7,5)	4680	0.0
(700,234,14,5)	2340	0.9
(435,187,15,5)	1073	0.0
(435,187,15,3)	1798	0.1
(435,187,7,5)	2324	0.0

즉, (treatment1) 중첩영역을 가로로 두 부분으로 분리하는 방법과 (treatment2) 중첩 영역을 세로로 두 부분으로 분리하는 방법, (treatment3) 중첩영역의 왼쪽아래 모서리를 포함하여 제거하는 방법 중 가장 좋은 해를 선택하는 과정을 거치는 것이다. 알고리즘의 자세한 구조는 참고논문 [2]을 참조하기 바란다. 본 연구를 위하여 fast algorithm은 발견적 기법(heuristic method)으로 고안되었으며, 본 알고리즘의 성능 평가를 위해 2차원 평면에서의 패턴을 생성하는 프로그램을 제작하여 패턴 생성 속도를 측정하였다. 본 프로그램은 팔렛의 길이 L , 폭을 W , 적재 대상물의 길이 l , 폭을 w 라고 할 때의 형태로 값이 입력되며, 그림 9는 (87,47,7,6)일 때의 출력값을 나타내고 있다.

Fast algorithm의 장점은 빠른 계산 시간에 있다. 표 1은 그림 9의 자동패턴생성 프로그램을 시뮬레이션 한 결과이다. 소요된 시간에서 확인할 수 있듯이, 대부분의 패턴을 1초 이내에 생성함을 알 수 있다.

위의 그림 9와 같이 사용자가 입력한 박스 및 팔렛의 크기에 맞는 최적의 적재 패턴이 생성되면, 패턴생성 프로그램은 각각의 대상물의 좌표 값을 계산하여 로봇 시뮬레이션 프로그램에 인가하게 된다.

III. 로봇시뮬레이션 프로그램(robot simulator)

산업용 로봇을 이용한 팔레타이징 작업을 실현하기 위하여 몇 가지 방법이 제안된 바 있다. 그 중 가장 일반적인 방법은 조작기(teach pendent)를 이용하여 사용자가 로봇의 예상 동작 궤적을 포인트 단위로 입력 시키고, 로봇이 입력된 좌표를 따라 반복 동작하도록 하는 것이었다. 또 다른 방법

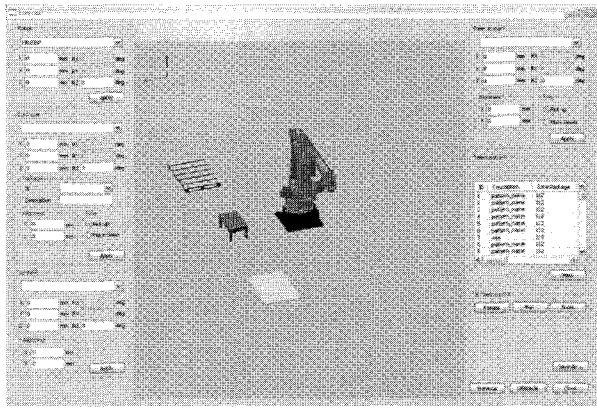


그림 10. 3차원 팔레타이징 로봇 시뮬레이터.
Fig. 10. Pattern generation program using the fast algorithm.

으로는 PC(Personal Computer)에서 로봇의 동작을 생성하고, 이를 통해 생성된 좌표 값을 로봇 제어기에 다운로드 하여 로봇이 구동되도록 하는 방법이 있다. 본 논문에서는 후자의 방법을 사용하되, 오프라인 소프트웨어(Off-line S/W) 내에서 로봇의 경로 생성을 사용자가 일일이 입력하는 방식이 아닌, 사용자가 임의로 구성된 작업 환경에 맞도록 실시간으로 경로를 생성하는 알고리즘을 도입하였다. 그림 10은 사용자 임의로 구성된 팔레타이징 작업 환경의 예를 나타내고 있으며, 여기에 추가로 사용자가 적재하고자 하는 작업물의 물리적 정보를 입력하면, 전 절에서 설명한 패턴 생성프로그램이 자동으로 패턴을 생성함으로써, 로봇이 작업물을 내려놓는 목표점을 생성한다.

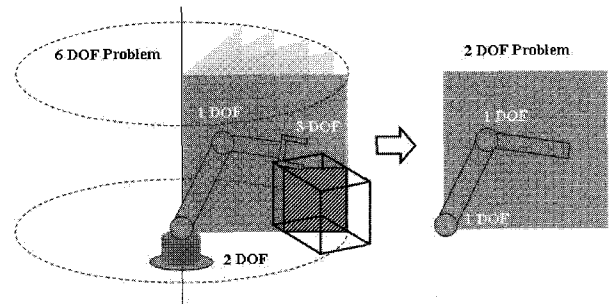
지금까지의 과정을 통해 팔레타이징 작업을 위한 로봇과 각 작업물의 위치가 정의되었고, 이러한 각 좌표를 경유하는 최적의 경로를 생성한 기법을 다음 장에 이어서 설명한다.

IV. 관절형상좌표 계(configuration space)와 A* 알고리즘의 적용

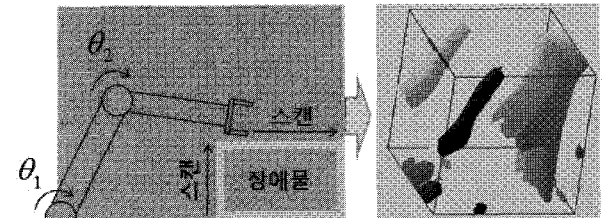
1. 자세 공간

본 절에서는 일반적으로 다 관절 로봇의 최적경로 생성을 위해서 많이 사용되는 체세 공간의 개념 및 경로탐색 알고리즘인 A* 알고리즘을 언급하고, 본 논문의 시스템을 그대로 적용한 A* 알고리즘의 시뮬레이션 결과 및 이를 보완하기 위하여 고안된 경로 생성기법인 'overlap method'를 제안한다. 자세 공간에서 자세란, 로봇 작동기(end effector)의 위치와 방향을 표시하기 위한 변수로 사용된다. 자세 공간이란 자세 값을 얻을 수 있는 모든 공간을 의미한다. 이러한 좌표 계에서 공간상의 각 점은 로봇의 작동기로 도달할 수 있는 공간상의 위치에 대한 로봇의 자세 값이다[3,4].

자세 공간상에서의 로봇의 자세 값을 정의하기 위해서 일반적으로 사용되는 방법은 평면분할기법(slice plane theory)이다. 이는 로봇의 기저 축을 중심으로 하여 로봇의 가동가능 범위를 평면으로 분할함과 동시에 말단부의 자유도를 고려하지 않음으로 인해서, 주요 링크의 동작 영역을 2차원 평면상에서 다룰 수 있는 기법이다. 또한 이러한 간략화를 통해서 로봇 주변의 장애물 역시 연속적인 2차원 평면의 결합으로서 처리할 수 있다. 그림 11(a)는 이러한 분할 평면 생성



(a) 분할 평면 생성



(b) 자세 공간에서의 장애물 생성

그림 11. 분할 기법을 통한 로봇 동작영역의 2차원화와 관절형상좌표 계에서의 장애물 재생성.

Fig. 11. 2D presentation of the robot operation area and regeneration of the obstacle in the configuration space using the slice plane method.

이론의 개념을 도시하고 있으며, 그림 11(b)는 2차원화된 장애물들을 새로운 좌표계인 관절형상좌표계(configuration space)의 기준으로 재 생성하는 모습을 나타내고 있다. 관절형상좌표계란 로봇의 말단부가 위치한 공간에 대한 각 관절의 좌표 값들을 일련의 좌표 값으로 표현한 좌표계로서, 로봇 매니퓰레이터가 장애물을 피해서 작업하는 최적 경로를 산출하기 위한 방법으로 널리 쓰이고 있다. 그림 11(b)의 우측 그림은 그림 11(a)의 장애물을 관절형상좌표 계의 기준으로 표현한 것이다.

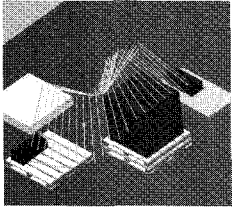
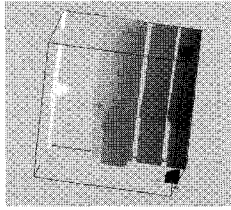
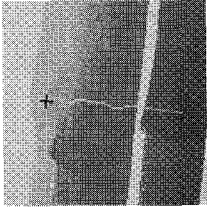
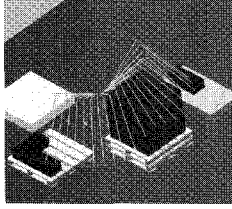
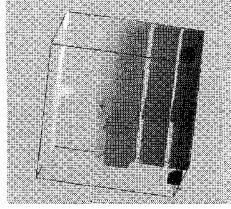
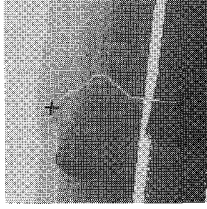
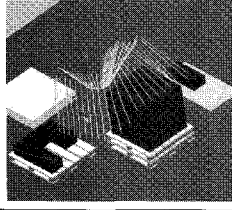
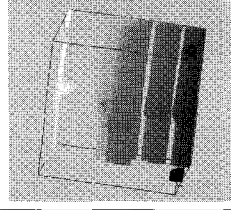
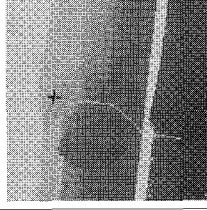
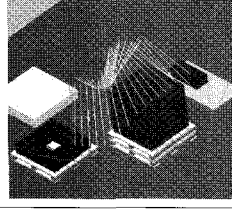
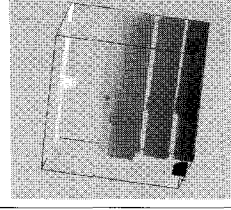
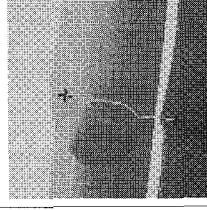
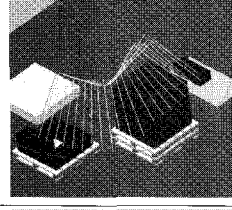
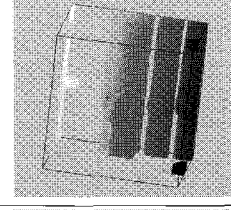
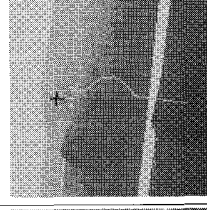
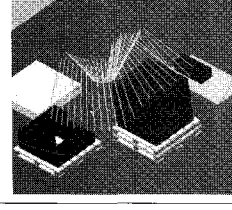
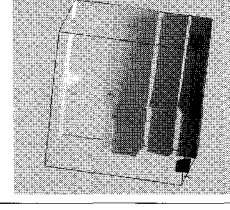
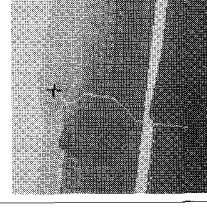
2. 경로생성을 위한 A* 알고리즘의 적용

A* 알고리즘은 자유 동작영역에서 로봇의 최적 경로를 생성하기 위한 강건한 계획법이다[5-7]. 본 연구에서는 서두에서 언급한 시작점-컨베이어벨트에서 공급되는 작업물의 위치-과 끝점-로봇이 작업물을 적재하는 위치- 사이의 최적경로를 생성하기 위해서 위의 관절형상좌표 계와 더불어 A* 알고리즘을 적용하였다. A* 알고리즘은 공간상에 사각형 격자를 형성하여 시작점으로부터 끝점까지의 이동거리를 인접 격자를 통해 이동할 수 있는 최단거리로 생성하는 기법이다.

이때 인접 격자에 위의 자세공간에서 생성한 장애물이 존재하면, 그 방향으로의 움직임은 불가능하다. 표 2는 임의 설정한 작업 환경 내에서 장애물을 스캔하고, 관절형상좌표 계로 표시한 후, A* algorithm을 이용하여 장애물을 회피하면서 작업물을 적재하는 작업을 시뮬레이션 한 결과의 일부를 도시하고 있다. 첫 번째 열은 로봇에 의한 적재 작업을 단계별로 도시한 것이며, 두 번째 열은 각 단계별로 스캔 한 로봇 주변 장애물을 관절형상좌표 계 상에 표시한 것이다. 3번째 열은 시작점(우측 붉은 x지점)에서 도착점(우측 푸른 x지점)까지 A* 알고리즘을 이용하여 검색하여 찾아낸 최적 경로를

표 2. A* 알고리즘을 이용한 팔레타이징 작업의 경로 생성 시뮬레이션.

Table 2. Trajectory generation simulation of the palletizing task using the A* algorithm.

Step	Task Layout	Configuration Space	Enlarged Image	Elapsed Time (sec)
1				3.132818
2				0.384636
3				12.267289
4				9.734590
5				0.981018
6				4.182875

나타내고 있다. 마지막 열은 각 단계별 연산 소요시간이다.

V. 로봇의 두께를 고려한 A* 알고리즘

전절에서 언급한 A* 알고리즘의 특징은 로봇의 체적을 고려하지 않는다는 것이다. 하지만 이는 실제적이지 않으며 시뮬레이션으로서의 가치가 현저하게 떨어진다. 따라서 본 연구에서는 분할 평면을 통한 공간스캔을 통해서 생성되는 장애물의 정보를 좀 더 실제적으로 얻어 보고자, 로봇의 체적을 고려한 평면 분할을 통해 A* 알고리즘을 적용하여 보았다.

1. 로봇 두께의 고려(1)

우선, 기존의 평면 분할 기법에서 두께가 없는 분할 평면

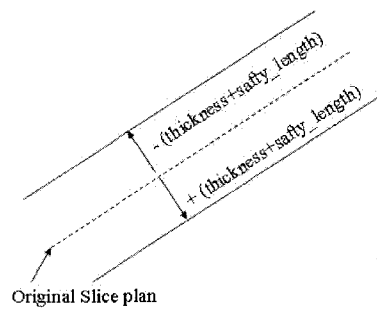


그림 12. 수정된 분할 평면.
Fig. 12. Modified slice plane.

이 아닌, 실제 로봇의 체적 및 두께를 고려한 분할 평면을 재구성하였다.

그림 12는 이러한 수정된 분할 평면을 도시하고 있다. 이러한 경우 분할 평면은 로봇팔의 양면을 고려한 안전거리 (thickness+safety length)의 두 배만큼의 두께로 새로이 정의하였다.

2. Convex Hull

두께를 가지는 분할평면을 이용하여 장애물을 스캔할 경우 기존의 3차원 장애물은 분할 평면상에 최대 4개의 꼭지점을 갖는 폐공간을 형성하는데 반해, 그림 13(a)와 같이 최대 12개의 꼭지점을 가질 수 있다. 이때, 로봇에 의한 장애물의 회피라는 측면에서 우리에게 중요한 정보는 최외곽에 형성되는 일련의 꼭지점 군이다. 따라서 convex hull이론을 사용하여 분할 평면상에 교점으로 생성된 꼭지점들 중 최 외곽의 점들을 연결하는 폐곡선을 생성하고, 이를 스캔 된 장애물의 형상으로 간주하는 기법을 사용하였다.

그림 14은 이러한 수정된 분할 평면의 효과를 도시하고 있다. 그림 14의 좌측은 체적이 없는 로봇 매니플레이터에 의

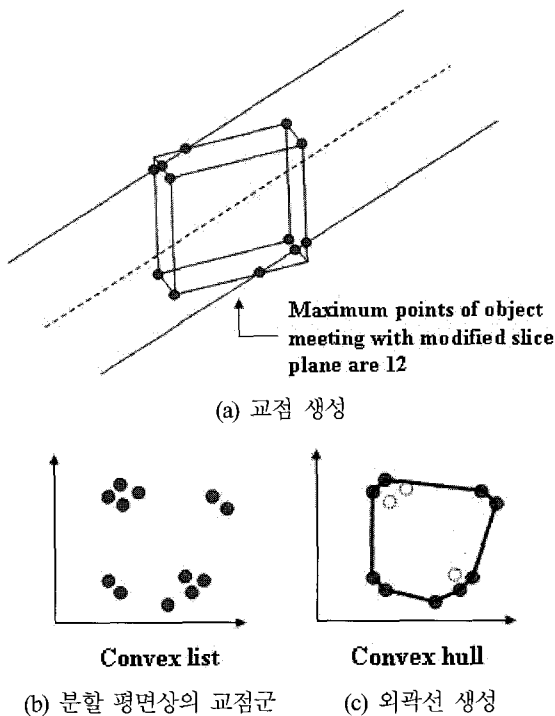


그림 13. Graham 법.

Fig. 13. Graham's method.

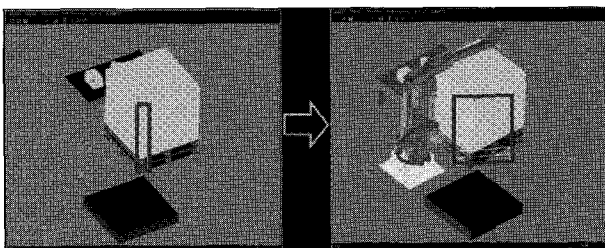


그림 14. 수정된 분할 평면의 효과.

Fig. 14. Effect of the modified slice plane.

해 스캔 된 장애물 단면이고, 그림 16의 우측은 체적을 지닌 로봇 매니플레이터에 의해 스캔 된 장애물 단면이다. 그림에서 유추할 수 있듯이, 체적이 없는 상태에서는 장애물과 충돌하지 않은 로봇 자세가 로봇의 체적을 반영하였을 경우, 충돌로 간주될 수 있다.

3. 로봇의 두께의 고려(11)

전 절에서 언급한 수평적 로봇 두께의 고려에 추가하여, 실제 3차원 공간상에서의 로봇 팔의 체적 및 로봇의 말단부에 장착되는 팔레타이징 작업 관련 작동기의 체적을 고려한 추가적인 처리가 필요하다. 그림 15은 본 연구에서 적용한 로봇의 수직 방향에 대해 설정한 물리적 경계조건을 도시하고 있다. 즉, 각각의 주요 암(arm)과 말단부(end-effector)의 길이와 형상을 고려하여 로봇의 형상을 체적이 없는 막대기(bar)가 아닌 일정 체적을 지닌 형상체로 재구성 한 것이다. 말단부는 본 시뮬레이션에서 적용한 박스 핸들링용 그리퍼 특정 모델의 최외곽선을 취하여 육면체로 형상화하였다.

4. 로봇의 두께를 고려한 경로 재생성

시뮬레이션 분석 결과, 로봇 몸체의 체적을 고려하였을 때와 그렇지 않을 때 -막대기로 간주- 의 시뮬레이션의 속도가 현저하게 차이 난다는 것을 확인할 수 있었다. 이는 본 연구의 대상인 로봇 팔레타이징 작업의 특성상 1회 단위작업마다 적재물이 적재되므로 다음 작업을 위해 로봇이 공간을 스캔 하였을 때 전 단계에서 적재된 기존 작업물로 인해 마치 장애물의 형상이 변경된 것으로 인식된다는 측면이 시뮬레이션에서 중요한 요소로 작용함을 의미한다. 이러한 이유로 인해, 로봇의 체적을 고려하였을 경우, A* 알고리즘의 시뮬레이션 결과는 작업 시간의 측면에서 불안정한 결과를 보였다(그림 20 참고). 다음절에서는 이러한 A* 알고리즘의 단점을 보완하기 위한 새로운 알고리즘을 제안한다.

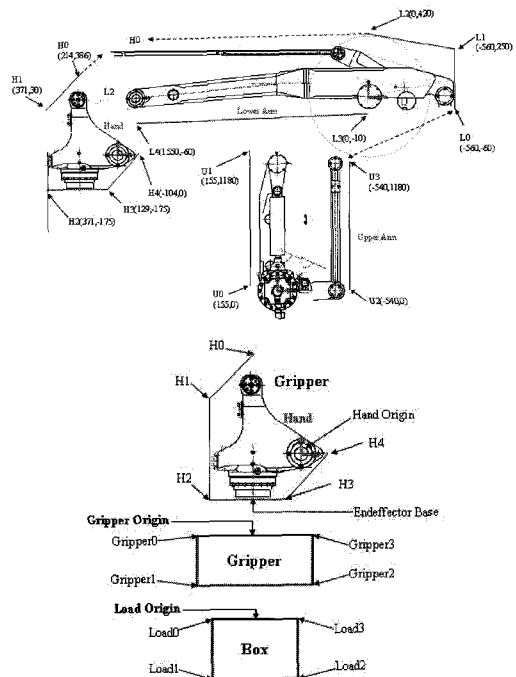


그림 15. 팔레타이징 로봇의 물리적 경계 설정.

Fig. 15. Definition of the physical boundary of the palletizing robot.

VI. Overlap Method

연산속도는 소프트웨어 개발의 측면에서 중요한 문제이다. 본 연구의 목적은 서론에서 언급하였듯이, 팔레타이징 자동화를 위한 OLP S/W를 개발하는 것이다. 표 3의 결과에서 알 수 있듯이 로봇의 실제 사이즈를 고려한 경로 생성 알고리즘의 경우, A* 알고리즘은 비효율적이다. 따라서 로봇 팔레타이징 작업의 특성을 고려한 새로운 경로 생성 기법을 제안하였다. 이는 기존의 경로 생성 기법이 가지고 있는 비효율성 - 매순간 스캔한 2차원 평면상의 장애물을 관절공간좌표계로 변환하여 A* 알고리즘을 사용하여 경로를 생성-을 개선할 목적으로 고안되었다. 본 연구에서 제안한 overlap method는 로봇 주변에 있는 작업물을 360 전방위로 1회 스캔한 뒤, 매순간 발생한 2차원 평면상의 장애물의 단면을 하나의 단면에 모두 중첩시켜, 하나의 장애물로 간주하여 최적화를 실행함으로써 기존 알고리즘의 비효율성을 개선하였다.

그림 16은 overlap method의 기본 원리를 도식화한 것이다. 그림 18(c), (d)와 같이 overlap method는 한 평면에 중첩된 장애물들의 최종 경계선만을 취득하여 로봇의 θ_2 와 θ_3 로 이루어진 관절형상좌표 계를 만들게 된다.

생성된 관절형상좌표 계는 로봇이 시작위치(pick up point)의 θ_1 각도에서 목표위치(place down point)의 θ_1 까지 이동하는 동안에 로봇이 θ_2, θ_3 각도를 움직여서 피해야 할 충돌영역을 표시하게 되고 (검은색 부분), (1)을 이용하여 로봇이 시작위치의 θ_2, θ_3 에서 충돌영역을 회피하여 목표위치의 θ_2, θ_3 에 도달하기 위해 거쳐야 할 최적의 θ_2, θ_3 (optimal via point)를 찾게 된다(그림 17). 여기서 θ_1 은 기저부 관절, θ_2 는 어깨관절, θ_3 는 팔꿈치 관절을 의미한다.

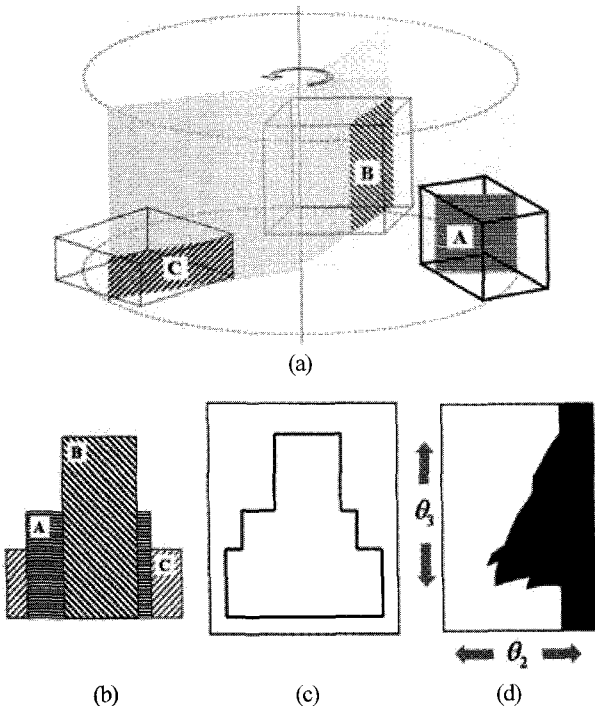


그림 16. Overlap method의 기본 개념.
Fig. 16. Basic concept of the overlap method.

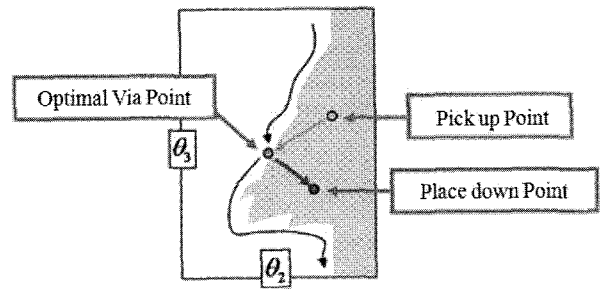


그림 17. 최적경로 생성을 위한 θ_2 와 θ_3 의 결정.
Fig. 17. Determination of θ_2 and θ_3 to generate trajectory.

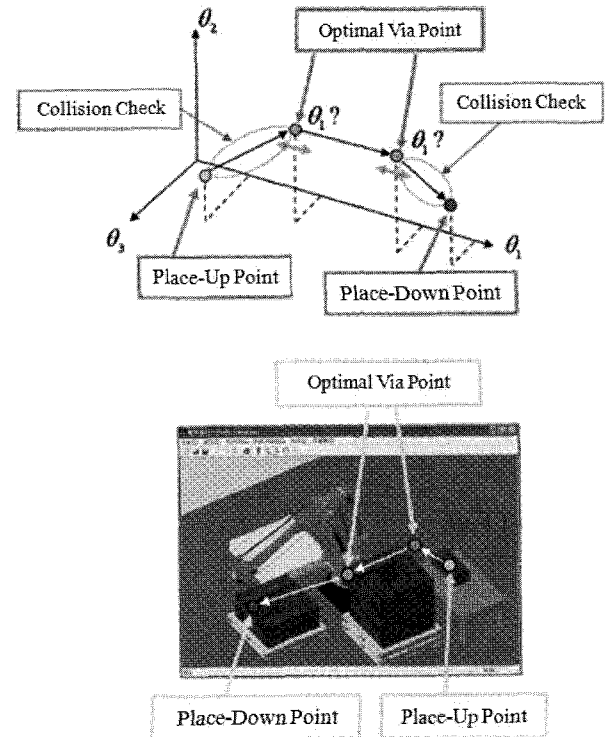


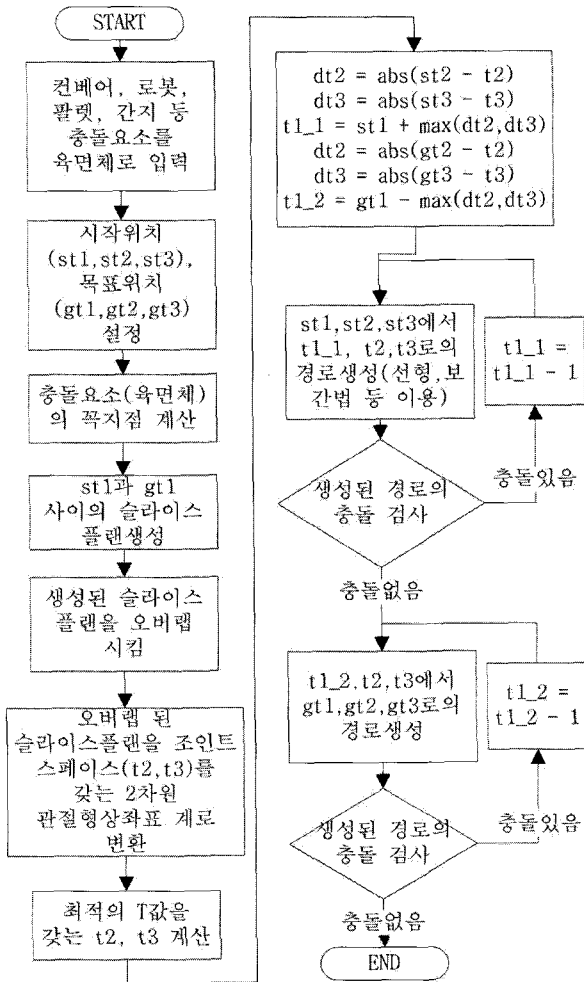
그림 18. 최적 경로점 생성을 위한 θ_1 의 결정.
Fig. 18. Determination of θ_1 to generate the optimal via point.

$$T_{opt} = A[abs\{(P_{via} - P_{pick-up})_{\theta_2}\} + abs\{(P_{place-down} - P_{via})_{\theta_2}\}] + B[abs\{(P_{via} - P_{pick-up})_{\theta_3}\} + abs\{(P_{place-down} - P_{via})_{\theta_3}\}] \quad (1)$$

θ_2, θ_3 가 정해지면, θ_1 을 변화시키면서 장애물과 로봇의 말단부의 충돌을 검사한다. 이러한 과정을 거쳐 첫 번째 경로점(optimal via point)을 획득한다. 첫 번째 경로점이 정해지면, 모든 장애물을 뛰어넘은 뒤 내려 놓는 위치로 접근하기 위해 말단부가 하강이 시작되는 하강 경로점에 이를 때까지 장애물과의 충돌을 검사하고, 두 번째 경로점에 이르면 하강을 시작하여 작업물을 정해진 위치에 내려 놓게 된다(그림 18). 그림 19는 overlap method의 알고리즘 구조를 도식하고 있다.

VII. 작업 시뮬레이션

본 논문에서는 지난 연구에서 제안한 적체물 패턴 생성 알



st1, st2, st3: 시작위치의 $\theta_1, \theta_2, \theta_3$
 gt1, gt2, gt3: 목표위치의 $\theta_1, \theta_2, \theta_3$
 t2, t3: 최적 경로점의 θ_2, θ_3
 t1_1: 첫 번째 최적 경로점의 θ_1
 t1_2: 두 번째 최적 경로점의 θ_1

그림 19. Overlap method의 알고리즘 구조.
 Fig. 19. Algorithm of overlap method.

고리즘 및 3차원 로봇시뮬레이터를 기반으로 하여 그것의 최적경로를 생성하기 위한 알고리즘을 제안하였다. 이는 기존의 분할 평면 기법 및 A* 알고리즘을 개선한 방법으로서 팔레타이징 작업에 초점을 맞춘 알고리즘이다.

표 3은 본 논문에서 언급한 알고리즘들의 성능을 평가하기 위해서 총 24개의 적재물을 이송 및 적재하는 작업을 설정하고, 로봇의 체적을 고려하지 않은 A* 알고리즘, 로봇의 체적을 고려한 A* 알고리즘 및 overlap method에 대한 각각의 단위작업(step)별 연산 소요시간을 나열한 것이다.

그림 20에 도시한 결과에서 확인할 수 있듯이 기존의 A* 알고리즘은 로봇의 체적을 고려하지 않았을 때에는 우수한 시뮬레이션 시간을 보이나, 로봇의 체적을 고려함과 동시에 시뮬레이션 성능이 극도로 불안정하고 느려짐을 알 수 있다.

표 3. A* 알고리즘, 수정 A* 알고리즘 및 장애물 및 overlap method의 시뮬레이션 결과.

Table 3. Results of the simulation of the A* algorithm, the modified A* algorithm, and the overlap method.

Step	Time (Sec.)		
	Line(A*)	Volume(A*)	Overlap Method
1	0.014051	2.380194	0.412106
2	0.01543	2.066007	0.427587
3	0.01558	1.857863	0.415959
4	0.017952	2.318304	0.43101
5	0.014286	2.265576	0.411975
6	0.016003	2.213547	0.422834
7	0.013669	1.360996	0.443131
8	0.014206	2.20403	0.440602
9	0.016555	1.328561	0.454023
10	0.015094	1.298407	0.438623
11	0.017387	1.660548	0.466195
12	0.01523	1.298854	0.4273
13	0.01889	0.886562	0.46002
14	0.01344	1.159015	0.428835
15	0.016348	1.091212	0.43728
16	0.01413	1.301314	0.424192
17	0.017107	0.386479	0.464915
18	0.016205	0.429178	0.47078
19	0.017836	0.361664	0.484123
20	0.014301	0.389356	0.439179
21	0.020215	0.278059	0.491373
22	0.014119	0.408476	0.441098
23	0.018028	0.288906	0.466881
24	0.014902	0.295928	0.439116

VIII. 결론

본 연구에서 제안한 overlap method는 수정 A* 알고리즘에 비해 현저하게 짧은 시뮬레이션 시간과 안정성을 보이고 있

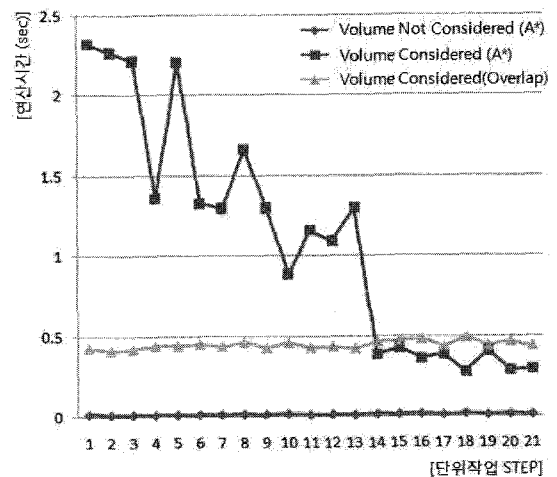


그림 20. 세 가지 경로 생성법에 대한 연산소요 시간 비교.
 Fig. 20. Comparison of the calculation times of the three methods.

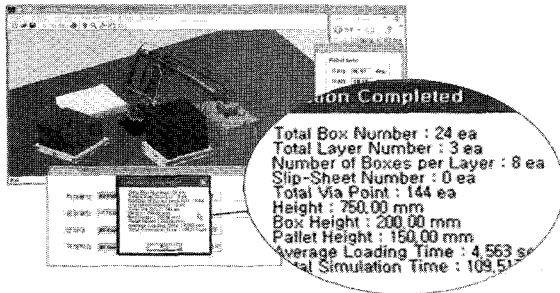


그림 21. Overlap method를 적용한 팔레타이징 작업 시물레이션 및 시물레이션 결과.

Fig. 21. Palletizing task simulation and the results of the use of the overlap method.

음을 확인할 수 있었다. 물론, 위의 시물레이션에서 14단계 이후에 연산 소요시간이 역전되면서 overlap method에 비해 수정 A* 알고리즘이 우수한 연산시간을 보이기도 하지만, 이는 적재가 진행될수록 다음 작업물을 내려놓는 높이-목표점의 절대높이가 올라가면서, 기존의 장애물들의 효과가 반감되는 현상에 기인한다.

따라서 본 연구에서 제안한 overlap method가 전체적으로 목표점의 높이 변화나 주변의 장애물의 영향을 받지 않고 일관되게 우수한 연산 결과를 도출함을 알 수 있다. 그림 21는 fast algorithm과 overlap method를 적용하여 개발한 OLP 팔레타이징 시물레이터의 프로토타입과 이를 이용한 작업 시물레이션의 산출 결과를 나타내고 있다. 본 시물레이터는 총 작업물의 개수 및 적재 높이, 작업 시간 외에 다양한 정보가 사용자에게 제공 된다.

IX. 고찰

본 연구는 생산 현장에서 최종 생산품의 적재를위해 사용되는 팔레타이징 시스템을 자동화 하기 위해 개발하고 있는 로봇 팔레타이저의 OLP 소프트웨어의 기존 작업을 소개하고, 팔레타이징용 로봇 소프트웨어의 경로 생성을 위해 적용한

A* 알고리즘에서 나타나는 연산 시간 측면의 문제점을 개선하기 위해 overlap method를 제안하였다. 본 연구에서 제안한 경로 생성 알고리즘은 팔레타이징 작업이라는 특정 작업에 기반하여 개발된 것이므로, 산업용 로봇을 사용하는 모든 공정에 활용하기는 어렵다. 다만, 로봇 팔레타이저를 위한 시물레이터로서는 우수한 성능을 이끌어 낼 수 있음을 확인하였다. 향후에는 본 논문에서 제안한 알고리즘의 성능 개선 이외에, 이러한 오프라인 소프트웨어를 실제 로봇 및 로봇 제어기와 연결하여 시물레이션과 실제 작업 사이의 유사성을 평가하고, 동시에 소프트웨어 인터페이스의 사용자 편의성을 개선하여 전체적인 시스템의 완성도를 향상시키는 것을 목표로 하고 있다.

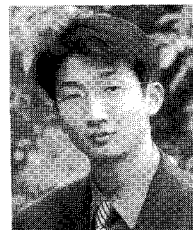
참고문헌

- [1] P. J. McKerrow, *Introduction to Robotics*, Addison-Wesley Ltd, p. 484, 1991.
- [2] G. Young-Gun. and M.-K. Kang, "A fast algorithm for two-dimensional pallet loading problems of large size," *European Journal of Operational Research* 2001, pp. 193-202, 2000.
- [3] Roy, Debanik, "Study on the configuration space based algorithmic path planning of industrial robots in an unstructured congested three-dimensional space: An approach using visibility map," *Journal of intelligent & robotic systems*, v. 43 no. 2/4, pp. 111-145, 2005.
- [4] J. H. Kim, J. S. Choi, H. Y. Kang, D. W. Kim, and S. M. Yang, "Collision-free path planning of articulated robot using configuration space," *In proceeding of Korean Society of Automotive Engineers*, vol. 2, no. 6, pp. 57-65, 1994.
- [5] C. W. Warren, "Fast path planning using modified A* method," *In proceeding of IEEE*, pp. 662-667, 1993.
- [6] T. L.-Perez, "A simple motion planning algorithm for general robot manipulators," *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 3, 1987.
- [7] C.W Warren, J. C. Danos, and B.W. Mooring, "An approach to manipulator path planning," *The International Journal of Robotics Research*, vol. 8, no. 5, pp. 87-95, 1989.



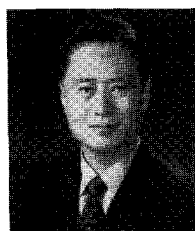
유승남

1978년 11월 11일생. 2004년 한양대학교 기계공학과(공학사). 2006년 한양대학교 정밀기계공학과(공학석사). 현재 한양대학교 기계공학과 박사과정 재학중. 관심분야는 로봇 기구학과 생체공학 및 4족 보행로봇.



임성진

1979년 10월 5일생. 2005년 한국항공대학교 항공기시스템공학과(공학사). 2007년 한양대학교 기계공학과(공학석사). 현재 한양대학교 기계공학과 박사과정 재학중. 관심분야는 로봇제어, 인공지능, 로봇 소프트웨어 플랫폼.



한창수

1955년 5월 1일생. 1983년 한양대학교 기계공학과(공학사). 1985년 University of Texas at Austin 기계공학과(공학석사). 1989년 University of Texas at Austin 기계공학과(공학박사). 현재 한양대학교 기계공학과 교수. 관심분야는 초정밀 로봇, 생체공학, 건설자동화로봇.



김성락

1961년 10월 18일생. 1984년 서울대학교 전기공학과(공학사). 1986년 KAIST 전기 및 전자공학과(공학석사). 1993년 KAIST 전기 및 전자공학과(공학박사). 현재 현대중공업(주) 기계전기연구소 로봇개발실장. 관심분야는 지강성 메커니즘의 정밀제어, 동역학 기반 궤적계획, 로봇 캘리브레이션, 네트워크 기반 로봇 제어 시스템.

로봇, 생체공학, 건설자동화로봇.