

신경망을 이용한 3 차원 잡는 점들의 해석적 결정

이현기#, 한창우*, 이상룡**

Analytic Determination of 3D Grasping points Using Neural Network

Hyun Ki Lee[#], Chang Woo Han^{*} and Sang Ryong Lee^{**}

ABSTRACT

This paper deals with the problem of synthesis of the 3-dimensional Grasp Planning. In previous studies the genetic algorithm has been used to find optimal grasping points, but it had a limitation such as the determination time of grasping points was so long. To overcome this limitation we proposed a new algorithm which employs the Neural Network. In the Neural network we chose input parameters based on the shape of the object and output parameters resulted from optimization with the GA method. In this study the GRNN method is employed, it has been trained by the result value of optimization method and it has been tested by known object. The algorithm is verified by computer simulation.

Key Words : Grasp Planning (잡기계획), Optimization (최적화), Neural Network (신경망)

1. 서론

인간은 물건을 보고 직관적으로, 경험적으로 물건을 잡게 된다. 하지만 로봇의 경우는 그렇지 못하다. 인간의 조작에 의해서나 계산에 의한 결과를 가지고 물건을 잡게 된다. 하지만 인간이 로봇을 조작할 수 없는 장소나, 물건에 대한 정보가 전혀 없는 공간에서 로봇이 작업해야 하는 상황이 된다면 계산적으로, 인간의 조작에 의한 작업을 할 수 없게 된다.

이런 문제들을 해결하기 위해 1980 년부터 잡기계획(Grasp Planning)이라는 제목으로 많은 연구들이 행하여져 왔다. 잡기계획에 있어서 가장 중요시되는 것은 손가락의 위치와 힘의 적용에 대한

안정성이다.

잡기계획에 있어서 가장 기초가 되고 중요한 토대가 되는 것은 Nyguen¹ 에 의해 소개되어진 "Force Closure"의 개념이다. 마찰을 고려하고, 물체와 그리퍼와의 접촉을 점 접촉으로 고려하여 안정적인 잡는 점(grasping points)을 찾는 방법을 제시하였다. Ferrari² 는 이런 "Force Closure"의 개념을 확대하여 "Grasp Quality"의 개념을 만들었다. 물체를 잡고있을 때 생기는 "wrench"들의 Convex Hull이 원점을 싸고있느냐의 여부로 잡기계획(Grasp Planning)의 안정성을 판정하였다.

근래의 연구는 3-손가락 이상의 다손가락(multi-finger)의 최적 잡는 점에 대한 연구가 많이 이루어 졌다. 다손가락은 2-손가락과는 다르게 2-

--- 2002년 9월 18일 접수
교신저자, 경북대학교 기계공학과
Email mradam1004@hotmail.com Tel. (053) 952-2229
* 영남이공대학
** 경북대학교 기계공학과

차원 해석이 불가능하여, 알고 있는 물체들을 이용하여 3 차원에서 해석을 수행하였다. Borst³ 는 "Grasp Quality"² 를 이용하여 빠르고, 강건한 잡는 점을 찾는 알고리즘을 제시하였고, Katada⁴ 는 리아프노프 안정도해석(Lyapunov Stability) 이론을 도입하여 안정한 잡는 점을 찾고, "Evolution Programming"을 이용하여 최적의 점을 찾는 알고리즘을 개발하였다. 또 유전자 알고리즘(Genetic Algorithm)을 이용하여 최적의 잡기계획을 구하는 방법도 있었다.⁵

하지만 이런 연구들의 결과는 최적적이고 안정적인 잡는 점(Grasping Points)를 발견 할 수 있었지만, 연산속도가 느리다는 단점을 가지고 있었다. 이런 단점을 보완하기 위해서 본 논문에서는 신경망을 이용하는 새로운 알고리즘을 제시하고자 한다. Fig. 1 은 본 논문에서 제시할 알고리즘의 순서를 간단하게 표현한 그림이다. 유전자 알고리즘의 결과를 이용하여 신경망을 학습시키고, 학습된 신경망을 이용하여 학습되었거나 되지 않은 물체의 최적 잡는 점을 찾아낸다.

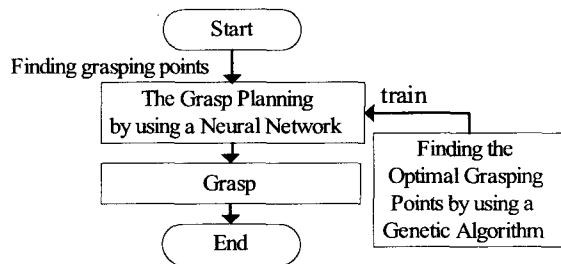


Fig. 1 Sequence of Proposed Algorithm

2. 3-손가락 핸드의 모델링

본 연구에서는 Fig. 2 와 같이 3-손가락 핸드를 모델링하여 모사실험을 수행하였다. 3-손가락 핸드는 다손가락에 해당하면서 해석이 가장 용이하고, 다손가락의 특성을 잘 표현할 수 있는 장점이 있다.

본 연구에서 사용된 3-손가락 핸드는 총 8 개의 회전운동을 하는 조인트로 구성된 8 자유도 핸드이다. 하지만 손가락을 각각 해석하면 손가락 1 은 2 자유도, 손가락 2 와 3 은 각각 3 자유도를 가진다. 위의 3-손가락 핸드에 대해서 기구학, 역기

구학, 자코비안 행렬, 손가락 끝단 힘에 대한 모델링을 하였다.

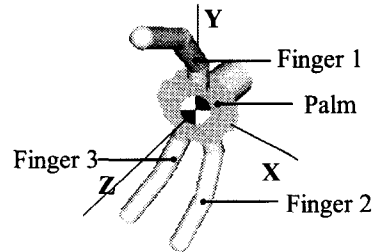


Fig. 2 Modeling of 3-finger hand

3. 유전자 알고리즘을 이용한 잡기계획

유전자 알고리즘을 이용하여 모델링 된 물체에 있어서 최적의 잡는 점을 이용하는 방법이다.⁷ 유전자 알고리즘의 결과를 이용하여 신경망을 학습 시키는 데이터로 이용한다.

강체로 가정된 임의의 물체에 위와 같이 힘이 작용한다면, 물체를 안정적으로 잡기 위해서는 힘의 평형이 맞아야 한다.

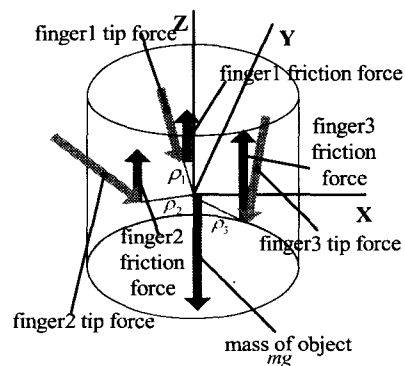


Fig. 3 Grasp Equilibrium

Fig. 3 에서 물체에 작용하는 모든 힘을 각각 f_1, f_2, \dots, f_n 이라 하고, 무게중심으로부터 힘이 작용하는 점까지의 위치 벡터를 각각 $\rho_1, \rho_2, \dots, \rho_n$ 이라 하면 식 (1)과 식 (2)와 같은 방정식을 만족하여야 한다.

$$\text{힘의 평형 : } \sum_{i=1}^n f_i = -mg \quad (1)$$

$$\text{모멘트의 평형 : } \sum_{i=1}^n \rho_i \times f_i = 0 \quad (2)$$

여기에서 m 은 물체의 질량, g 는 중력 가속도를 가리킨다.

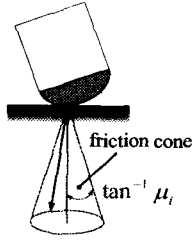


Fig. 4 Coulomb Friction Model

또, Fig. 4 와 같이 물체를 들기 위해서는 손가락 끝단과 물체 사이에 마찰력이 작용해야 한다.

$$|n_i \times f_i| \leq \mu_i (f_i \cdot n_i), \quad i = 1, \dots, n \quad (3)$$

여기에서 n_i 는 물체의 표면에 수직인 단위 벡터이고, μ_i 는 마찰계수를 가리킨다.

최적화 알고리즘을 적용하기 전에 먼저 목적 함수와 제한조건을 정의한다. 목적함수는 아래와 같이 정의할 수 있다.

$$F = D(p) \quad (4)$$

여기에서 $D(p) = \sum_{i=0}^n (p_i)^2$ 이고, n 은 손가락의 갯수를, p_i 는 각 손가락에서 손가락 끝단에 작용하는 힘으로 정의한다. 식(7)은 물체를 잡을 때 힘이 가장 적게드는 위치를 찾는다는 것을 의미한다. 제한조건은 Grasp Equilibrium 조건을 이용하였다.

위의 조건들에 의해 최적화해야할 파라미터들은 아래와 같다.

- ① 손가락 1,2,3 에 대한 표면점의 높이
- ② 손가락 2,3 에 대한 표면점의 각도
- ③ 각 조인트에 작용하는 토크
- ④ 3-손가락 핸드의 높이와 각도

위의 식들은 비선형 방정식으로 표현이 되고, 위의 최적화 파라미터를 가지고 아래와 같은 수식으로 표현할 수 있다.

$$\text{Minimize } F(y) \quad (5)$$

$$\text{Subject to } g_i(y) \leq 0, i = 1, \dots, n \quad (6)$$

여기에서 l 은 제한조건의 개수이다. 식(5)와 식(6)을 풀기 위해 식(5)와 식(6)을 벌칙 함수로 변형시켰다.

$$L(y) = F(y) + P \sum_{i=1}^l (g_i^+(y))^2 \quad (7)$$

위의 식(7)에서 $F(y)$ 는 목적함수, P 는 벌칙값 (penalty value)으로 충분히 큰 값을 선택하고⁸, $g_i^+(y) = \max\{0, g_i(y)\}$ 는 i 번째 제한조건에 대한 위반의 크기를 나타낸다.

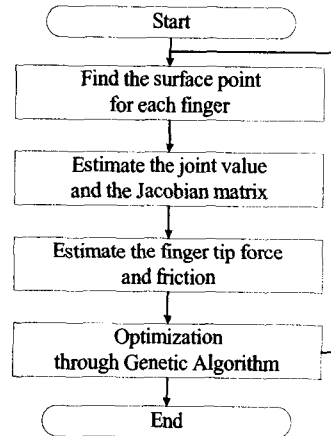


Fig. 5 Flow chart of optimization

식(7)에서 최적의 값을 찾기 위해서 본 연구에서는 Fig. 5 와 같은 순서로 최적화를 하였고, 최적화를 위해 유전자 알고리즘을 사용하였다.

4. 신경망을 이용한 잡기계획

신경망을 적용하기 전에 먼저 신경망의 학습이 필요하다. 신경망의 결과로 얻어져야 하는 것은 유전자 알고리즘을 이용하였을 경우와 동일하다. 입력 파라미터는 모델링 된 물체의 형상과 관계가 있다. 따라서 본 연구에서는 물체에서 형상을 정의 하기 위해 아래와 같은 입력 파라미터들을 설정 하였다.

입력의 첫번째와 두번째 파라미터는 모델링된 물체의 최고, 최저 깊이의 값으로 모델링된 물체의 높이와 연관이 있다. 이 파라미터는 3-손가락 핸드의 높이와 손가락들의 표면점의 높이와 연관이 있다. 나머지 파라미터는 물체의 단면의 모습을 구하기 위한 파라미터들로 Fig. 6 과 같이 설정 하였다.

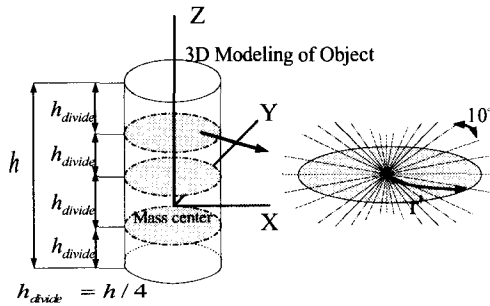


Fig. 6 Input parameters of Neural Network

물체의 높이를 세부분으로 나누고, 각 영역에서 각 10 도 마다의 반지름의 길이를 입력 파라미터로 사용하였다. 결과적으로 110 개의 입력 파라미터와 24 개의 출력 파라미터를 신경망에 적용하였다. 이때 24 개의 출력 파라미터들은 최적화 방법에 있어서의 최적화 대상 파라미터들을 적용하였다.

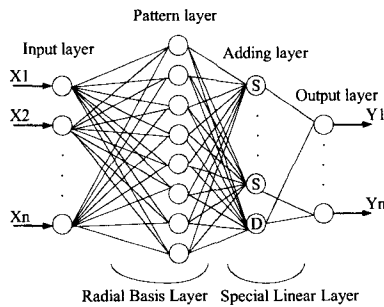


Fig. 7 GRNN

다음으로 신경망을 설계하여야 한다. 본 연구에서는 아래의 그림과 같은 GRNN(Generalized Regression Neural Networks)⁶ 모델을 선정 하였다. 이 GRNN 모델은 예측 모델로서 학습이 용이하고, 값을 예측하는 곳에 있어서 뛰어난 기능을 가지고

있다. GRNN 모델은 아래의 그림과 같이 4 개의 층으로 이루어져 있다. Pattern 층은 입력층과 연결이 되어 있고 각 입력패턴에 관해 하나의 뉴런을 가진다. Adding 층은 출력의 개수보다 한 개 많은 뉴런을 가지고 있다. 그림에서 S-adding 뉴런은 합하는 요소이고, D-adding 뉴런은 나누는 요소로 한 개가 존재한다.

5. 모사실험과 결과

물체에 대한 학습이 필요하다. 학습을 위해서 알고 있는 물체에 대해서 유전자 알고리즘을 통해 입력과 출력에 관한 결과를 얻어 내었다. Fig. 8 은 학습을 위해 선택한 다양한 물체의 모양이다. 본 연구에서는 원통형 물체에 대해서만 학습을 실시하였다. 이것은 일반적인 물체의 경우 원통형 물체가 많고, 원통형의 적용후에 일반적인 물체에도 적용이 가능함을 보이기 위함이다. 아래의 원통형 물체들은 높이의 차이, 원통의 반지름의 차이, 타원의 크기의 차이, 각도의 차이를 가지고 있다. 모두 99 개의 학습 데이터를 구하였다.

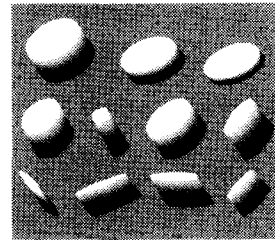


Fig. 8 Objects to get the training data

Fig. 9 는 위의 학습을 위한 물체 중에서 높이가 45mm 이고, 지름이 60mm 인 물체에 대한 학습의 결과를 보여 주고 있다. Fig. 9 (a)는 모사 실험의 결과를 보여 주고 있고, Fig 9 (b)는 GA 에서 각 반복 횟수에 따른 최적의 값을 볼 수 있다. 10 번 미전에서 아주 최적의 값에 아주 근접해 감을 볼 수 있다.

위에서 구해진 물체들을 이용하고 최적화 알고리즘을 이용하여 학습을 위한 데이터들을 구하였다. 먼저 최적화 알고리즘을 적용하기 전에 파라미터들을 셋팅하여야 한다. 물체와 손가락 끝단과의 마찰 계수는 1.0 으로, 물체의 밀도는 일정한

것으로 보았고, 3-손가락 핸드의 각 링크의 길이는 50mm로 설정하였고, 목적함수의 벌칙 값은 적당히 큰 값으로 $1.0e+20$ 으로 설정하였다. 위의 형상 중에서 한가지를 선택하여 최적화의 결과를 살펴 보면 아래와 같다.



Fig. 9(a) Optimization Result

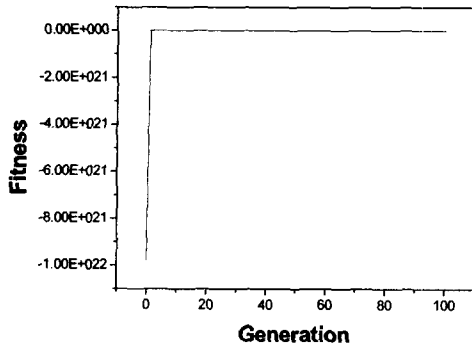


Fig. 9(b) Fitness Value of Optimization

Fig. 9 The optimization result of Grasp Planning

Table 1 The result of Optimization parameters

| variables | palm | Surface point | | |
|----------------|---------|---------------|---------|---------|
| | | Finger1 | Finger2 | Finger3 |
| Angle (degree) | -45.696 | 90.000 | 194.514 | 240.669 |
| Height (mm) | 114.335 | 30.077 | 33.016 | 34.5327 |

| variables | | Finger1 | Finger2 | Finger3 |
|----------------|--------|---------|---------|---------|
| Angle (degree) | Joint0 | | 73.669 | 67.952 |
| | Joint1 | 68.608 | 37.747 | 46.353 |
| | Joint2 | 102.529 | 141.439 | 140.975 |
| Torque (g mm) | Joint0 | | -0.736 | 0.428 |
| | Joint1 | 70.341 | 125.009 | 120.166 |
| | Joint2 | 3.204 | 31.311 | 22.0987 |

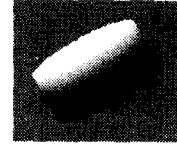


Fig. 10(a) Shape

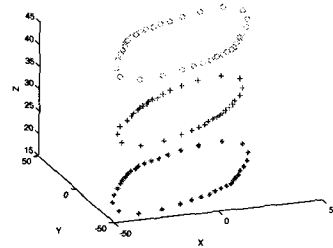


Fig. 10(b) Input Parameters

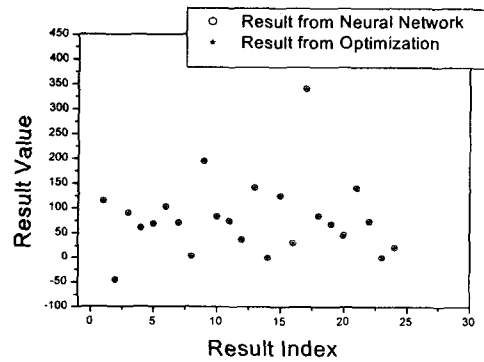


Fig. 10(c) Comparing the result from between neural network and optimization

Fig. 10 The input shape and parameters for test to train the network and result

신경망을 학습한 후에 학습에 사용되지 않은 물체를 이용하여 신경망이 바르게 학습이 되었는지 확인하여 보았다.

Table 1 은 최적화의 결과 값들을 볼 수 있다. 이것으로 학습에 의한 결과와 최적화에 의한 결과를 비교해 볼 수 있다. Fig. 10(a)의 물체는 높이 60mm, 45 degree 로 기울어져 있고, 장축의 길이는 110.40mm, 단축의 길이는 36.15mm 의 타원의 단면을 가지고 있는 물체로 학습에서는 사용되지 않은 물체이다. Fig. 10 (b)는 단면 3부분의 형상 파라미터인 입력 파라미터를 보여준다. Fig. 10 (c)에서 볼 수 있듯이 유전자 알고리즘을 이용하여 구한

출력 파라미터들과 신경망을 이용하여 구한 결과와는 거의 차이가 없음을 볼 수 있고, 실제적으로 5%미만의 오차를 가지고 있었다. Table 2 에서 그 결과를 볼 수 있듯이 Table 1 과 거의 유사한 결과를 볼 수 있다.

Table 2 The result of Neural Network

| Variables | Palm | Surface point | | |
|----------------|---------|---------------|---------|---------|
| | | Finger1 | Finger2 | Finger3 |
| Angle (degree) | -46.723 | 90.000 | 194.514 | 241.554 |
| Height (mm) | 114.452 | 29.247 | 30.436 | 34.355 |

| Variables | | Finger1 | Finger2 | Finger3 |
|----------------|--------|---------|---------|---------|
| Angle (degree) | Joint0 | | 73.669 | 67.952 |
| | Joint1 | 68.608 | 37.747 | 46.353 |
| | Joint2 | 102.529 | 141.439 | 140.975 |
| Torque (g mm) | Joint0 | | -0.633 | 0.524 |
| | Joint1 | 71.156 | 126.279 | 121.155 |
| | Joint2 | 3.203 | 32.299 | 21.184 |

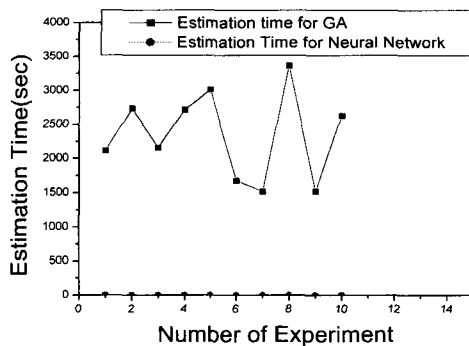


Fig. 11 The compare of the estimation time between using GA method and Neural Network method

Fig. 11 은 최적화 방법을 이용했을 경우와 신경망(Neural Network)을 이용했을 경우에 있어서 계산 시간을 비교한 그림이다. 각기 다른 물체 10 개에 대해서 실험을 수행 하였다. 최적화의 경우 30 분에서 40 분정도의 계산 시간이 소요되지만 신경망(Neural Network)의 경우 1 초나 2 초정도의 계산 시간이 소요 되었다.

6. 결론

본 논문에서 제시한 알고리즘은 기존 최적화의 결과와의 오차는 0.1% 미만의 차이를 가지면서, 정확도에는 차이가 없고, 연산 시간에 있어서는 상당한 개선을 가져 왔다. 기존의 유전자 알고리즘을 이용한 방법은 연산시간이 15 분정도 소요 되었으나 본 연구에서 제시한 방법은 2 초 정도의 시간이 소요 되었다. 기존의 유전자 알고리즘에서는 볼 수 없는 한번 학습후에 다시 그 물체를 만났을 경우 빠른 시간안에 최적화의 결과를 찾을 수 있었다.

하지만 학습을 위한 데이터가 많이 필요하고, 시간이 많이 소요 된다는 단점이 있다. 하지만 실시간으로 학습을 수행할 수 있는 기능을 갖춘다면 실제 작업환경 하에서도 적용이 가능하리라 본다.

참고문헌

1. V.-D. Nguyen, "Constructing force-closure grasps," *The Intl. J. of Robotics Research*, 7(3), 1988.
2. C. Ferrari and J. Canny, "Planning optimal grasps," *In Proceeding of the IEEE Intl. Conf. on Robotics & Automation*, Nice, France, pp. 2290-2295, 1992.
3. Ch. Borst, M. Fischer, and G. Hirzinger, "A Fast and Robust Grasp Planner for Arbitrary 3D Objects," *In Proc. of the IEEE Intl. Conf. on Robotics & Automation*, Detroit, Michigan, 1999.
4. Y. Katada, M. Svinin, K. Ohkura, and K. Ueda, "Optimization of Stable Grasps by Evolutionary Programming," *In Proc. of the 32nd ISR*, 2001.
5. Hyun-Ki Lee, Myun-Hee Kim and Sang-Ryong Lee, "Optimization of 3D Grasping Points with Whole 3D Modeling for Unknown Object," *In Proc. Of the 2nd China-Korea Joint Workshop on Robotics*, Shenyang, China, 2002.
6. Hyun-Hyup Lee and Kyung-II Mun, "Fuzzy-Neuro by using a MATLAB," *Ajin Press*.
7. 박강, "능동적 원격감시를 위한 스테레오 카메라 시스템의 개발," *한국정밀공학회 추계학술대회*, 2001.
8. Ashok D. Belegundu and Trupathi R. Chandrupatla, "Optimization Concepts and applications in engineering," *Prentice Hall*.