

유전자 알고리즘을 이용한 원기둥형 물체의 자세 추정 방법

정규원*

(논문접수일 2005. 1. 19, 심사완료일 2005. 3. 11)

Pose Estimation of a Cylindrical Object Using Genetic Algorithm

Jeong, Kyuwon*

Abstract

The cylindrical object are widely used as mechanical parts in the manufacturing process. In order to handling those objects using a robot or an automated machine automatically, the pose of the object must be known. The pose can be described by two rotation angles; one angle about the x axis and the other about the y axis. In the many previous researches these angles were obtained by the computationally intensive algorithm, that is, fitting the data as a polynomial and doing pseudo inverse. So that, this method required high performance microprocessor. In this paper in order to avoid complex computation, a new method based on a genetic algorithm is proposed and analyzed through a series of simulations. This algorithm utilized the geometry of the cylindrical shape. The simulation results show that this method find the pose angles very well in most cases, but the computation time is randomly changed because the genetic algorithm is basically one of the random search method.

Key Words : Cylindrical Object(원기둥형 물체), Pose Angles(자세 각도), Robot(로봇), Genetic algorithm(유전자 알고리즘), Population(모집단), Crossover rate(교배율), Mutation rate(돌연변이율)

1. 서 론

여러 가지 제품의 생산 공정에서 원기둥형으로 되어 있는 부품이 많이 있으며, 이들을 로봇등을 이용하여 자동으로 취급하는 경우가 많이 있다. 또한 각종 사회 간접자본에

사용되는 상수도, 송유관, 하수도등에 대형 파이프가 많이 사용되고 있다. 이들 부품의 취급을 자동화 하거나 파이프 등을 자동으로 검사하기 위해서는 원기둥형 물체의 중심방향을 알아야 할 필요가 있다. 그러나 이들 파이프 또는 원기둥형 물체는 주행 방향의 기준으로 사용할 수 있는 모서리

* 충북대학교 기계공학부 (jeong@chungbuk.ac.kr)
주소: 361-763 충북 청주시 흥덕구 개신동 12

또는 꼭지점이 없기 때문에 로봇이 주행할 때 주행 방향을 유지하기 어렵다. 그러나 산업용 로봇에는 원기둥형 물체의 표면의 좌표를 구하기 위한 센서가 있거나, 취급을 자동화하기 위한 센서가 부착되어 있는 경우가 많으므로 이로부터 구한 좌표를 이용하여 로봇 몸체 기준좌표 또는 그리퍼의 방향을 기준으로 물체의 중심축 방향을 찾고자 한다. 이렇게 구해진 방향은 이동형 검사용 로봇을 유도하는데 사용될 뿐만 아니라, 물체의 자세에 맞추어 작업해야 되는 보수 작업 등에도 사용될 수 있으며, 검사 데이터를 기록할 때에도 유용하게 사용된다. 또한 부품의 핸들링 또는 용접작업등에도 유용하게 적용될 수 있다^(1,2). 참고문헌^(3,4,5)에서는 물체로부터 얻은 좌표를 최소자승법을 이용하여 곡면 맞춤을 하고 이를 이용하여 중심방향을 구하였는데, 이때는 곡면 맞춤에 또는 의사역행렬을 구하기 위하여 많은 시간이 들거나, 성능이 좋은 마이크로 프로세서를 이용하여야 한다. 따라서 본 논문에서는 범용 저가의 마이크로 프로세서를 이용하여 원기둥형 물체의 중심방향을 구하기 위하여 유전자 알고리즘을 이용하는 방법을 제시하고 시뮬레이션을 통하여 그 타당성을 보이고자 한다.

2. 자세 추정 방법

2.1 알고리즘의 이론적 배경

본 논문에서 사용되는 거리센서는 로봇 또는 측정기 등에 부착되어 있으므로 최초로 얻어지는 데이터는 로봇 또는 측정기를 기준으로 한 데이터이다. 그러나 사용자 또는 로봇을 구동하기 위해서는 물체의 좌표가 사용자를 기준으로 한 좌표계로 주어져야 한다. 따라서 이 데이터를 이용하기 위해서는 측정기를 기준으로 구한 데이터를 원기둥 물체에 정의된 사용자 좌표계로 변환하여야 한다. Fig. 1을 이용하여 설명하면 다음과 같다. 로봇 또는 측정기의 좌표계 원점 O_R 와 물체에 부착된 사용자 좌표계의 원점 O_P 의 위치가 다르나, 일차적으로 측정된 데이터를 사용자가 편리하게 이해할 수 있고, 다음 절에서 설명하게 될 것과 같이 일반적인 경우에도 본 논문의 알고리즘은 사용될 수 있기 때문에 두개의 원점은 일치한다고 가정한다. Fig. 1에 보인 것과 같이 측정기의 중심에 좌표계 $O_R(x_R, y_R, z_R)$ 을 부여하고, 사용자가 정의한 물체의 좌표계 $O_P(x_P, y_P, z_P)$ 가 있으며, 물체의 중심축 방향이 y_P 와 일치하는 방향으로 하며, 이는 로봇의 방향 y_R 과는 z_R 을 중심축으로 $-\alpha$, y_R-x_R 평면 즉 x_R 축을 중심으로 β 만큼 회전된 방향에 있다고 가정한다. 먼저 거리 측정용 센서를 사용하여 물체 벽면의 점 $P(x_R, y_R, z_R)$ 를 구

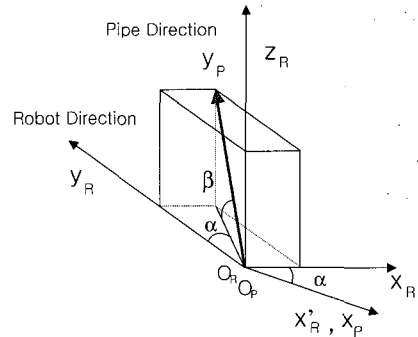


Fig. 1 Reference Coord. for posture description

하고, 측정용 센서에 부여된 기준점부터 물체 벽면의 한 점은 다음의 벡터로 표시된다.

$$r = x_R i + y_R j + z_R k \quad (1)$$

거리 측정용 센서를 물체주위로 회전시키면서 물체의 벽면을 주사하여 다수의 점의 좌표를 구한다.

Fig. 2를 이용하여 본 논문의 알고리즘을 설명하면 다음과 같다. 원기둥형 물체 표면의 점 P들을 원기둥 중심축 방향 즉 y_R 방향에서 보면 모든 점 P들은 원주상에 있게 된다. 그러나 중심축과 다른 방향 즉 y_R 에서 보면 이들 점들이 타원 궤적 상에 있게 된다. 따라서 물체를 보는 방향을 변경시키면서 모든 점들이 하나의 원주상에 오게 하면, 이때의 O_R 좌표축과 이루는 각도가 물체의 방향이 된다. 이 각도를 구하기 위하여 여러 가지 방법이 있겠으나, 본 논문에서는 저가의 마이크로프로세서를 사용하는 것을 목표로 하여 계산량을 적게 할 수 있는 방법을 사용하려고 한다. 이 방법 중에 가장 많이 사용되는 것이 유전자 알고리즘인 것으로 생각된다. 따라서 다음 절에서 간략히 설명하는 유전자 알고리즘을 이용하고자 한다.

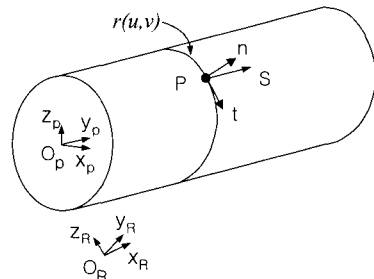


Fig. 2 Principal direction and the related vectors

2.2 유전자 알고리즘을 이용한 방법

1) 이론

앞에서 언급한 바와 같이 거리 데이터를 곡면으로 맞추며 관의 형상인 원기둥 곡면을 얻고 이를 이용하여 관의 방향을 구하는 방법을 사용할 수 있다. 그러나 이 방법은 곡선 맞춤을 하는데 계산량이 많고, 과정이 복잡하기 때문에 처리 시간이 많이 필요하며 컴퓨터의 성능이 우수하여야 한다. 따라서 본 논문에서는 이러한 문제점을 극복하기 위하여 유전자 알고리즘을 사용하여 계산시간을 단축하고 처리과정을 간략히 한다. 전체적인 처리과정은 Fig. 3에 보인 것과 같다.

물체의 방향은 Fig. 1과 같이 α 와 β 의 각도로 주어져 있다고 가정하고, 임의의 α 와 β 를 선정한다. 초기에는 임의로 선정하고 이들을 이용하여 앞에서 언급한 거리데이터를 다음 식 (2)와 (3)을 이용하여 좌표변환한다.

$${}^P R_R = R(x_p, -\beta)R(z_p, \alpha) \quad (2)$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = {}^P R_R \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (3)$$

위의 식들에서 $R(x_p, -\beta)$ 는 x_p 기준으로 $-\beta$ 만큼 회전을 나타내며, $R(z_p, \alpha)$ 는 z_p 축을 기준으로 α 만큼의 회전을 나타낸다. $[x \ y \ z]_p^T$ 는 가정된 물체를 기준으로 계산된 물체 벽면의 좌표이다. 따라서 이들은 물체의 중심축인 yp 방향에서 보았을 때 물체의 직경을 갖는 원주상에 있어야 한다. 변환된 좌표축이 물체의 중심축과 일치하지 않으면 이들은 원주상에 있지 않고 흐트러지게 된다. 따라서 이와 같은 특성을 이용하여 주어진 데이터들이 모두 하나의 원주상에 있을 때의 회전각도 α, β 구한다. 본 논문에서는 이들 각도를 선정할 때에 서론에서 언급한 바와 같이 어려운 계산을 생략하기 위하여 Fig. 3과 같이 반복적인 계산을 이용하는 유전자 알고리즘을 사용한다.

새로운 세대의 포플레이션을 생성할 때는 가장 적합도가 좋은 유전자를 선택하는 엘리티즘과, 적합도가 높은 유전자가 더욱 많이 선택되도록 하는 룰렛 휠(roulette wheel) 기법을 사용하며, 크로스오버 기법을 이용하여 새로운 유전자를 만들고, 지역해에 빠지는 경우를 방지하는 뮤테이션기법을 사용한다⁽⁶⁾.

본 논문에서 대상으로 하고 있는 원기둥형 물체에 대한 적합도는 측정된 데이터들의 중심의 위치, 지름 또는 반지름 등의 여러 가지를 이용하여 표현할 수 있는데, 본 논문에서는 원기둥형 물체를 대상으로 하고 있으며, 이들은 대부분

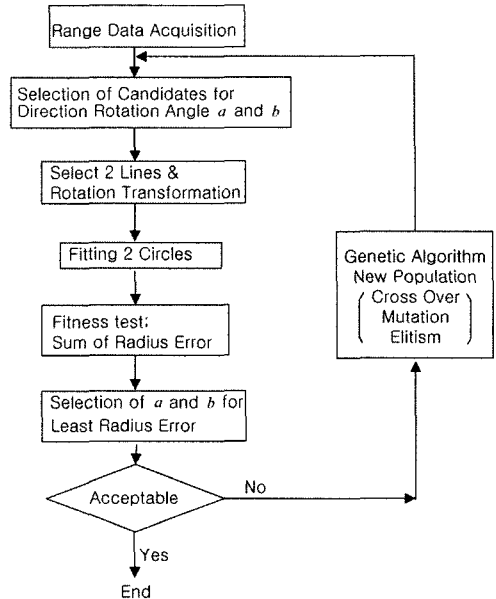


Fig. 3 Procedure for overall algorithm

설계단계에서 지름 또는 반지름이 결정되어 있으므로 좌표 변환된 거리 데이터가 미리 주어진 물체의 반경과의 오차가 적은 것을 적합도가 우수한 것으로 하였다.

이 문제는 다음과 같이 볼 수 있다. 임의로 선택된 α, β 각도로 좌표변환을 하여 측정된 데이터를 변환한다. 이들 좌표에서 임의의 3개의 점을 선택하여 이들을 지나는 원의 반경을 구한다. 측정된 점 데이터들이 노이즈와 측정오차들을 포함하고 있으므로 이와 같은 원을 5개를 구하여 각각의 반경 $r_i, i = 1 \sim 5$ 을 구한다. 이 원의 반경을 이용하여 적합도함수는 다음과 같이 정한다.

$$fitness = \frac{1}{\sum_{i=1}^5 (r_0 - r_i)} \quad (4)$$

여기에서 r_0 는 설계상에 주어진 물체의 반경이다. 이와 같은 적합도를 이용하여 유전자 알고리즘에서 새로운 α, β 를 반복하여 선정한다. 이 과정에서 다음의 반경오차 r_e 가 일정한 이하가 되면 알고리즘을 종료한다.

$$r_e = \sum_{i=1}^5 (r_0 - r_i) \quad (5)$$

유전자 알고리즘은 여러 변수, 즉 포플레이션 개수, 유전자의 길이, 교배율, 돌연변이율, 엘리티즘의 유전자수 등에

따라 수렴속도와 정밀도등이 영향을 받기 때문에 이들의 영향을 살펴볼 필요가 있다. 이들에 대한 결과는 다음 절에 시뮬레이션 결과에서 자세히 설명하였다.

2) 시뮬레이션 결과

다음의 그림들 Fig. 4 ~ Fig. 6은 본 논문의 아이디어의 타당성을 보여주기 위하여, 알고리즘의 진행에 따른 중간 결과들을 보여주고 있다. Fig. 4는 거리센서를 사용하여 원기둥형 물체의 측면의 좌표를 측정된 것을 3차원적으로 보여주고 있다. 이 그림에서 보듯이 측정방향이 물체의 방향과 일치하지 않기 때문에 두개의 서로 떨어진 선으로 보인다. Fig. 5에서는 이 데이터를 이용하여 앞에서 설명한 바와 같은 전체적인 과정을 거쳐 유전자를 이용하여 계산된 회전각도 α 와 β 를 변화시켰을 때 계산된 실측 좌표들의 반경과 미리 주어진 물체의 반경의 차이가 세대의 변천 즉 알고리즘의 반복회수에 따라 변화되어 가는 과정을 보여주고 있다. 문제의 특성상 반경의 차이가 적을수록 더욱 좋은 유전자가 선택된 것으로 간주한다. 이 그림에서는 15세대 정도에서 수렴하고 있는 것을 보여주고 있으나, 유전자 알고리즘의 랜덤한 특성 때문에 포플레이션의 수, 교배율, 돌연변이율등에 따라 변화가 많이 있다. 이는 다음의 시뮬레이션 결과에서 논의 할 것이다.

Fig. 6은 최종적으로 얻은 회전각도를 이용하여 Fig. 4의 데이터를 좌표변환 했을 때 z_p 방향에서 본 데이터의 분포이다. 이 그림에서 보듯이 측정된 데이터들이 하나의 원주상에 있는 것을 볼 수 있다. 이 그림에서는 실측된 데이터들을 사용했기 때문에 완전한 원주상에 있지 않고 오차를 갖고 있는 점들이 있으나 센서의 오차와 물체 표면의 불균일등에 기인한 오차가 있기 때문인 것으로 생각된다. 이때의 α 와 β

값이 측정기에 상대적인 물체의 방향임을 알 수 있으며, 현실적으로 로봇의 유도, 물체의 취급 등에 사용하는데 충분하다.

Fig. 7 ~ Fig. 10은 유전자 알고리즘의 변수들 즉 포플레이션 개수, 유전자의 길이, 교배율, 돌연변이율에 따른 영향을 비교해 보기위하여 시뮬레이션한 결과들이다. 이들 그림에서 각각의 시뮬레이션은 포플레이션 개수는 10개, 20개 30개인 경우, 또한 유전자길이는 16비트인 경우에 대하여 교배율이 0.1, 0.5, 0.9인 경우와, 돌연변이율 0.1, 0.5, 0.9인 경우의 조합으로 아홉 가지의 경우에 대하여 시뮬레이션 하였으며, 또한 유전자 길이에 대한 영향을 보기 위하여 Fig. 10에서 유전자길이를 8비트로 하였을 때 결과를 보여주며, 이는 Fig. 7과 비교할 수 있다. 유전자 알고리즘은 기본적으로 확률을 이용하기 때문에 통계적인 분포를 얻기 위하여 시뮬레이션을 각각의 조건에 따라 100회씩 하였으며, 각각의 조건에서 일정한 조건에 도달할 때까지의 시간으로 비교하였다.

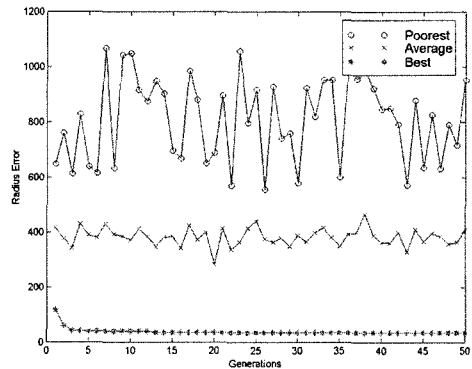


Fig. 5 Fitness(radius error) variation with the population generation

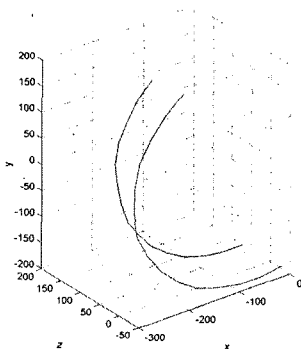


Fig. 4 The range data from the mobile robot to the pipe surface

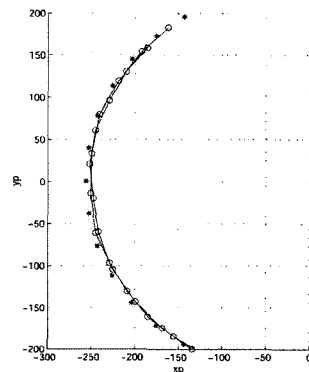


Fig. 6 The aligned range data to the pipe axis

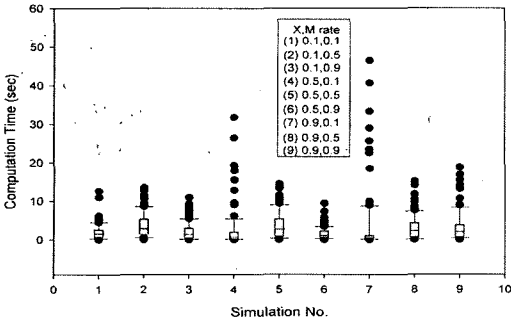


Fig. 7 알고리즘의 변수에 따른 계산시간의 변화(포플레이션수 : 10, 유전자의 길이 : 16bit, x :교배율, M : 돌연변이율)

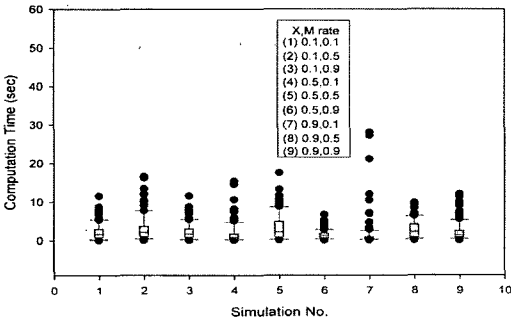


Fig. 8 알고리즘의 변수에 따른 계산시간의 변화(포플레이션수 :20, 유전자의 길이 : 16bit, x : 교배율, M : 돌연변이율)

계산시간은 개별 시뮬레이션에서 최상의 적합도가 일정한 값으로 수렴할 때까지의 걸린 시간인데, 이는 각 조건에서 세대의 변천 횟수에 비례한다고 할 수 있다. 각 시뮬레이션에서 사각형 박스는 계산결과 25% ~ 75%내의 계산회수를 보여주며, 짧은 가로 직선은 10% ~ 90%범위의 계산회수를 보여주며, 원형 점으로 표시된 부분은 이들 범위 밖의 아웃라이어 (outlier)들이다.

따라서 Fig. 7의 경우 시뮬레이션 4번과 7번의 경우 아웃라이어가 많으며, 다시 말하면 안정된 적합도를 얻기까지 세대변천이 많이 있었다는 것을 의미한다. 반면에 시뮬레이션 3번과 6번의 경우 아웃라이어가 적으므로 적은 세대변천으로도 좋은 결과를 얻은 것으로 볼 수 있다.

Fig. 8 ~ Fig. 10의 경우도 같은 방법으로 계산결과를 정리한 것이며, Fig. 7은 포플레이션 수가 10인 경우, Fig. 8은 20인 경우, Fig. 9는 30인 경우이며, 이들 모두 유전자의 길이는 16bit를 사용하였다. 유전자의 길이가 길면 하나의 유전

자에 대응하는 회전각도가 매우 정밀해지며, 포플레이션수가 많아지면 더욱 넓은 범위에서 해를 찾게 된다. 따라서 일반적으로 포플레이션수가 많아지면 짧은 시간(세대변천) 내에 해를 구할 수 있을 것으로 기대된다. 이는 Fig. 7 ~ Fig. 9에서도 전체적인 경향이 나타남을 볼 수 있다. 유전자의 길이가 길면 정밀한 결과를 얻을 수 있지만 반면에 계산시간이 길어지므로 가급적이면 실용적인 범위 내에서 짧을수록 효과적일 수 있다. Fig. 10에서는 포플레이션수와 유전자의 길이를 작게 했을 경우의 계산 결과를 보여주고 있는데, Fig. 9의 경우보다 약간 안 좋아졌다. 따라서 현재는 Fig. 9의 시뮬레이션 4번 또는 6번의 경우가 가장 좋은 결과를 주고 있다.

위의 결과들에서 아웃라이어가 많이 나타나는 경우에는 세대 변천 중에 포플레이션을 초기화 하는 방법, 교배율 또는 돌연변이율등을 변화시키는 방법등을 사용하면 이런 경우를 방지할 수 있을 것으로 생각된다.

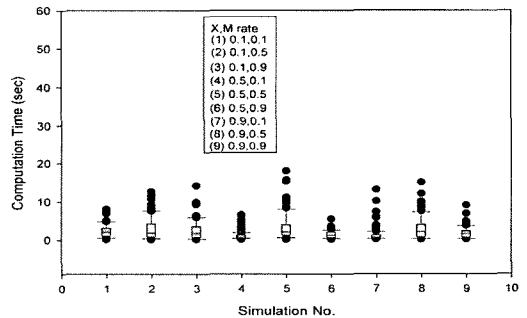


Fig. 9 알고리즘의 변수에 따른 계산시간의 변화(포플레이션수 : 30, 유전자의 길이 : 16bit, x :교배율, M : 돌연변이율)

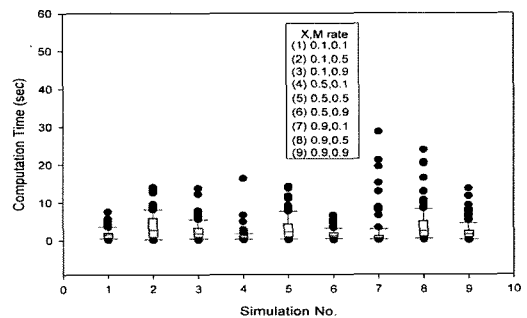


Fig. 10 알고리즘의 변수에 따른 계산시간의 변화(포플레이션수 : 10, 유전자의 길이 : 8bit, x : 교배율, M : 돌연변이율)

3. 결 론

본 논문에서는 거리 측정용 센서를 이용하여 취득한 좌표 데이터를 이용하여 원기둥형 물체의 자세를 인식하는 방법을 개발하였다. 기존의 개발된 방법과는 달리 어려운 계산이 필요하지 않도록 원기둥형 물체의 기하학적인 특성을 이용하여 유전자 알고리즘을 사용하는 방법으로서, 회전각도변환에 필요한 삼각함수는 테이블 룩업 방법을 이용하여 간단히 프로그래밍할 수 있으므로 저가의 마이크로 프로세서를 이용할 수 있을 것으로 기대된다. 실측된 데이터를 이용하여 알고리즘을 시뮬레이션한 결과 대부분의 경우에는 짧은 시간 즉 적은 세대변천 내에서 실용적인 결과가 얻어지나, 앞에서 언급한 바와 같이 오랜 세대 변천 후에 수렴하는 경우가 발생하는 경우가 있는데 이와 같은 경우에는 초기화를 다시 하거나 교배율등을 바꾸는 방법으로 해결할 수 있을 것으로 생각되며 추후 더욱 연구를 해야 할 것이다.

후 기

이 논문은 2004년도 충북대학교 학술연구지원사업의 연구비 지원에 의하여 연구되었음.

참 고 문 헌

- (1) Joo, K., 1998, "Unoccluded Cylindrical Object Pose Measurement Using Least Square Method," *J. of the KSPE*, Vol. 15, No. 7, pp. 167~174.
- (2) Chae, H. C., 2002, "Development of a 3D Shape Construction Software Using Unorganized Point Data," *Trans. of KSMTE*, Vol. 11, No. 1, pp. 1~9.
- (3) Lee, B. S., 1999, *A Study on the Pipe Defect Inspection Method using a laser Slit Beam Sensor*, Chungbuk National Univ. Dept. of Industrial Education, M.S. Thesis.
- (4) Jeong, K. W., 2003, "Pose Estimation of a Cylindrical Object for an Inspection Robot," *Trans. of KSMTE*, Vol. 12, No. 1, pp. 8~15.
- (5) Faux, I. D., and Pratt, M. J., 1979, *Computational Geometry for Design and Manufacture*, Ellis Horwood.
- (6) Jang, J.S.R., Sun, C. T., and Mizutani, E., 1997, *Neuro-Fuzzy and Soft Computing*, Prentice-Hall International Inc.