

인공신경망을 이용한 삼차원 물체의 인식과 정확한 자세계산

박 강*

(1998년 11월 11일 접수)

3D Object Recognition and Accurate Pose Calculation Using a Neural Network

Kang Park

Key Words: 3D Object Recognition(삼차원 물체 인식), Object Pose(물체의 자세), Centroidal Profile(圖心輪廓), Correspondence(대응관계), Neural Network(인공신경망), CAD(전산 기원용설계)

Abstract

This paper presents a neural network approach, which was named PRONET, to 3D object recognition and pose calculation. 3D objects are represented using a set of centroidal profile patterns that describe the boundary of the 2D views taken from evenly distributed view points. PRONET consists of the training stage and the execution stage. In the training stage, a three-layer feed-forward neural network is trained with the centroidal profile patterns using an error back-propagation method. In the execution stage, by matching a centroidal profile pattern of the given image with the best fitting centroidal profile pattern using the neural network, the identity and approximate orientation of the real object, such as a workpiece in arbitrary pose, are obtained. In the matching procedure, line-to-line correspondence between image features and 3D CAD features are also obtained. An iterative model posing method then calculates the more exact pose of the object based on initial orientation and correspondence.

1. 서론

생산 자동화에 대한 관심이 고조되면서 기계에 의한 물체의 인식에 관한 연구가 활발히 진행되어져 왔다. 그러나 물체 인식 시스템이 생산 시스템에 사용되기 위해서는 물체인식 속도가 빨라야하고, 인식의 성공률이 높아야하고, 또한 계산된 물체의 자세가 정확해야한다. 이런 조건들을 만족시키기 위해서 대부분의 기존 물체인식 시스템에서는 그 대상이 이차원 물체⁽¹⁾에 국한되었다. 빠른 인식속도와 안정성을 위하여 인공신경망을 물체인식에 이용하려는 노력들도 있어왔는데, 그

적용분야 역시 글자,⁽²⁾ 비행기,⁽³⁾ 그리고 밝은 바탕 위에 놓여진 평면적인 물체^(4,5) 와 같은 이차원 물체에 국한되었다. 인공신경망을 삼차원 물체의 인식에 적용하려는 노력이⁽⁶⁻⁸⁾ 있었으나, 이들은 물체의 형상에 제약이 있고 계산된 물체의 자세의 정확도가 낮아서 진정한 의미의 삼차원 물체의 인식으로 보기에는 무리가 있었다.

삼차원 물체의 인식에 있어서 가장 큰 문제는 꼭지점이나 모서리 같은 삼차원 CAD 모델의 특징(Feature)들과 이차원 입력 이미지의 특징 사이의 대응관계(Correspondence)를 어떻게 찾아낼 것인가 하는 점이다. 삼차원 CAD 모델과 이차원 입력 이미지는 서로 차원이 틀리기 때문에 이차원 물체의 인식에서와 같이 모델 특징과 입력 이미지의 특징을 직접 매칭을 할 수가 없다. 따라서 이 대응관계를 찾기 위해서 먼저 대응관계를

* 회원, 명지대학교 기계공학과

가정한 후 觀點(Viewpoint)을 계산해내고, 이를 이용하여 삼차원 모델을 이차원 이미지로 투사하여 앞의 가정의 진위를 확인하는 '가정과 확인'(Hypothesis-and-verification) 방법⁹⁾이 쓰이기도 한다. 그러나, 물체의 형상이 복잡해질수록 가능한 가정의 수가 기하급수적으로 늘어나서 물체인식 시간이 폭발적으로 늘어나는 단점이 있다.

삼차원 형상의 복잡 정도에 관계없는 빠른 물체인식과 정확한 자세계산을 위하여 이 논문은 인공신경망을 이용한 삼차원 물체의 인식 시스템(Centroidal profile and neural network based 3D object recognition and localization system: PRONET)을 제안한다. PRONET은 인공신경망의 병렬연산 성질을 이용하여 삼차원 물체를 빨리 인식하며, 또한 물체의 초기 자세를 계산한 후 반복계산을 이용하여 그 자세를 미세 조절함으로써 정확한 자세를 얻을 수 있다.

2. PRONET의 개관

Fig. 1은 PRONET 시스템의 개관을 나타낸다. PRONET은 크게 학습단계와 실행단계의 두 단계로 나뉘어진다. 학습단계에서 PRONET은 앞에서 언급한 '가정과 확인' 방법을 피하기 위하여 삼차원 CAD 모델을 한정된 갯수의 觀點에서 바라봤을 때 생기는 이차원 이미지들의 집합으로 표시하는 多觀點모델을 이용한다. 공간상에 고루 분포한 觀點을 얻기 위해 표면이 80개의 정삼각형으로 만들어진 정다면체(Tessellated sphere)를 정의하고, 각 삼각형의 중심을 '모델 觀點'으로 정의한다. 삼차원 CAD 모델을 정다면체의 중심에 놓고 각 모델 觀點에서 정다면체의 중심으로 바라보면 그 물체의 이차원 이미지가 얻어진다. 그로부터 물체의 윤곽을 추출해낸 후, 윤곽선상에 일정한 간격으로 데이터 점들을 추출하고, 윤곽의 圖心으로부터 데이터 점들까지의 거리를 저장함으로써 圖心輪廓(Centroidal profile) 패턴을 얻는다. 이 과정에서 삼차원 CAD 모델의 형상 특징과 이차원 이미지의 특징간의 대응관계가 추출되어서 도심윤곽 패턴에 저장된다. 圖心輪廓는 이차원 패턴형태로 주어지며 3계층 인공신경망에 오차역전파(Error back propagation)방법을 이용하여 학습된다. 학습이 끝난 인공신경망은 실행단계에서 이용되기 위하여 저장된다.

실행단계에서는 삼차원 물체를 카메라가 임의의 觀點에서 바라봄으로써 '입력이미지가' 얻어진다. 물체의 이미지는 색상 정보를 이용하여 배경으로부터 분리되고, 이로부터 '입력 도심윤곽 패턴'을 추출한다. '입력 도심윤곽 패턴'은 저장되어 있던 인공신경망들에 차례로 입력되고, 출력으로서 물체의 종류와 입력관점의 근사치가 얻어진다. 즉, 미지의 '입력관점'에서 얻어진 입력 이미지가 주어졌을 때, 多觀點모델에 사용되었던 80개의 '모델관점' 중에서 입력관점과 가장 비슷한 관점을 찾아내고 이를 입력관점의 근사치로 한다. 이 때 삼차원 물체의 형상 특징과 이차원 이미지의 특징사이의 대응관계도 얻어지는데, 이 대응관계와 관점의 근사치를 반복적 모델 자세계산 방법에 입력하여 물체의 정확한 자세를 구할 수 있다.

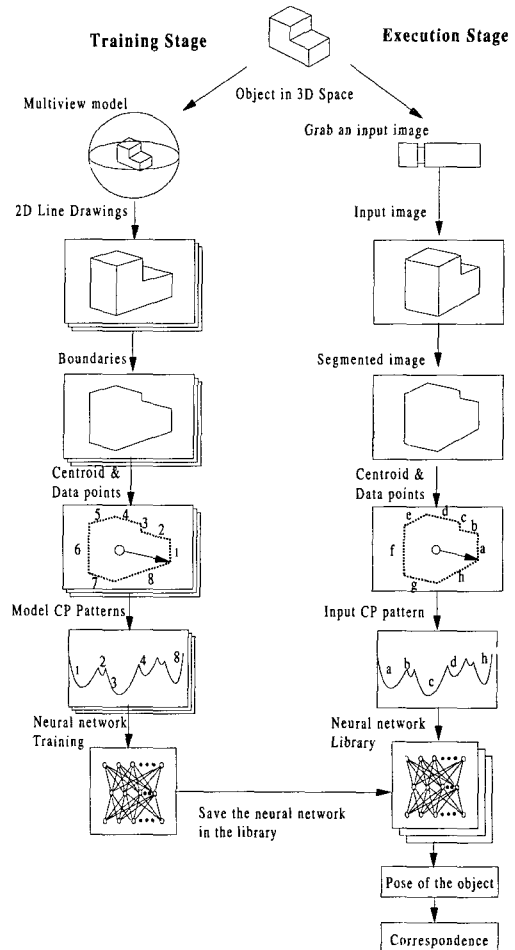


Fig. 1 Overview of PRONET system.

3. Multiview 모델

주어진 관점에서 물체의 윤곽이 어떻게 보일 것인가를 알아내기 위하여 먼저 물체를 CAD 솔리드 모델러로 모델링한다. 물체 주위로 균일하게 분포한 유한한 수의 관점들을 얻기 위하여 正多面球⁽¹⁰⁾(Tessellated sphere)를 사용한다. Fig. 2는 각각 80 개와 180개의 정삼각형 면을 갖는 정다면구를 나타내는데, 이 논문에서는 관점의 수를 줄이기 위하여 80개의 삼각형 면을 갖는 것을 사용한다. 물체를 정다면구의 중심에 서로의 좌표축이 일치되도록 위치시킨 후, 각 삼각형 면의 중심에서 정다면구의 중심으로 단위벡터를 정의함으로써 관점을 정의할 수 있다. Fig. 3에서 관점은 단위구의 표면상의 점 P를 모델좌표계 {M}에 대하여 나타낸 mP 로 정의 된다.

$${}^mP = \{x, y, z\}^T \text{ where } x^2 + y^2 + z^2 = 1 \quad (1)$$

카메라 좌표계(C)는 mP 점에서 구의 중심으로의 방향인 접근벡터 (\vec{a})와 구면에 접하면서 아래로 향하는 방향벡터(\vec{o})를 정의함으로써 구해진다. 노말벡터(\vec{n})는 $\vec{o} \times \vec{a}$ 로 주어진다.

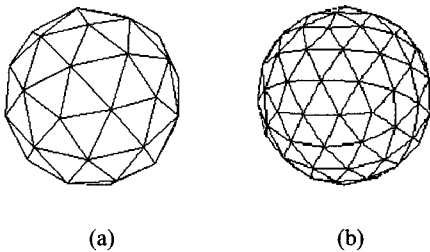


Fig. 2 Tessellated spheres with (a) 80 facets and (b) 180 facets.

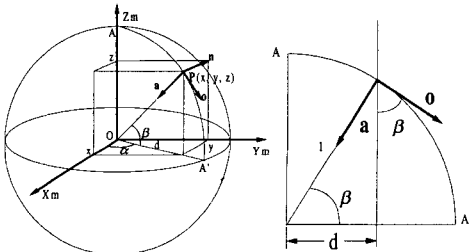


Fig. 3 The definition of a viewpoint and its rotation matrix.

$$\text{즉, } x = d \cos \alpha, \quad y = d \sin \alpha, \quad z = \sin \beta, \\ d = \sqrt{(x^2 + y^2)} \text{ 일 때}$$

$$\vec{a} = \begin{bmatrix} -x \\ -y \\ -z \end{bmatrix},$$

$$\vec{o} = \begin{bmatrix} \sin \beta \cos \alpha \\ \sin \beta \sin \alpha \\ -\cos \beta \end{bmatrix} = \begin{bmatrix} xz/d \\ zy/d \\ -d \end{bmatrix},$$

$$\vec{n} = \vec{o} \times \vec{a} = \begin{bmatrix} -y/d \\ x/d \\ 0 \end{bmatrix} \quad \text{여기서 } d \neq 0 \quad (2)$$

로 주어지며, $d=0$ 인 경우, 즉 관점이 $(0, 0, 1)$ 또는 $(0, 0, -1)$ 인 경우 $\vec{o} = \{1, 0, 0\}^T$ 과 $\vec{o} = \{-1, 0, 0\}^T$ 를 각각 사용한다.

mP 와 cP 를 각각 모델과 카메라 좌표계에 의하여 표시되는 P점의 좌표라고 하였을 때 카메라 좌표계에서 모델좌표계로 변환시키는 회전 행렬(cR)은 ${}^mP = {}^cR {}^cP$ 의 관계를 만족하며 다음과 같이 주어진다.

$${}^cR = \begin{bmatrix} \vec{n} & \vec{o} & \vec{a} \end{bmatrix} \\ = \begin{bmatrix} -y/d & xz/d & -x \\ x/d & zy/d & -y \\ 0 & -d & -z \end{bmatrix} \quad \text{여기서 } d \neq 0 \quad (4)$$

주어진 관점에서 CAD 모델을 봤을 때 모델의 자세가 카메라에 대하여 어떻게 표현될 것인지를 알아내기 위해서 cR 의 역행렬인 mR 이 필요하다.

$${}^mR = \begin{bmatrix} -y/d & xz/d & 0 \\ xz/d & zy/d & -d \\ -x & -y & -z \end{bmatrix} \quad \text{여기서 } d \neq 0 \quad (5)$$

관점들의 좌표 (x, y, z) 가 정다면구로부터 정해졌을 때 회전 행렬 cR 의 요소들은 식 (5)로부터 실수 값으로 주어진다.

$${}^cR = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (6)$$

Euler 각도 (ϕ, θ, ψ) 가 주어졌을 때 회전 행렬은 다음과 같이 주어지므로,

$$R = R_{z,\phi} R_{x,\theta} R_{z,\psi}$$

$$= \begin{bmatrix} C\phi C\psi - S\phi C\theta S\psi & -C\phi S\psi - S\phi C\theta C\psi & S\phi S\theta \\ S\phi C\psi + C\phi C\theta S\psi & -S\phi S\psi + S\phi C\theta C\psi & -C\phi S\theta \\ S\theta S\psi & S\theta C\psi & C\theta \end{bmatrix} \quad (7)$$

${}^c_m R$ 의 ϕ , θ , 그리고 ψ 는 식 (6)과 (7)을 이용하여 구할 수 있다.

각 관점마다 이 Euler 각도가 저장되어 실행 단계에서 물체의 자세를 계산하는데 사용된다. PRONET 多觀點 모델은 위에서 구한 관점들에서 물체를 바라봤을 때 생기는 이차원 이미지들로부터 도심윤곽패턴을 얻음으로써 얻어진다.

4. 모델도심윤곽패턴의 생성

물체가 솔리드 모델러를 이용하여 모델링된 후, 이 모델로부터 모델도심윤곽패턴을 생성하는 순서는 Fig. 4에서 예시된 바와 같이 다음 네 단계로 나뉘어진다.

(1) 곡면을 다면체로 근사시키기 위하여 삼차원 CAD 모델의 표면을 작은 단면으로 나눈다. (Fig. 4(a))

(2) 80개의 모델 관점들에서 바라본 이차원 이미지들을 계산해내고 은선을 제거한다. (Fig. 4(b))

(3) 물체의 윤곽을 이용하여 도심과 윤곽을 계산한다. 인공지능망은 정해진 숫자의 데이터 세트가 입력되어야 하므로, 윤곽선상에 정해진 숫자만큼의 데이터 점들을 추출한다. (Fig. 4(c)) 이 논문에서는 100개의 데이터점들이 사용되었다.

(4) 데이터점들 중에서 도심으로부터 가장 먼 점으로부터 시작하여 도심과 데이터점 사이의 거리를 연속적으로 저장함으로써 도심윤곽패턴을 얻어낸다. (Fig. 4(d))

도심윤곽패턴이 인공지능망에 사용되기, 위해서는 정해진 관점에서 본 물체의 이미지가 스크린 상에서 평행/회전이동과 확대/축소가 일어나더라도 동일한 도심윤곽패턴이 얻어져야 한다.

도심윤곽패턴은 도심과 윤곽선사이의 상대적 거리로 이루어지므로 물체의 이미지가 모니터 스크린 상에서 평행이동과 회전이동을 하더라도 변하지 않는다. 그러나 물체와 카메라간의 거리가 가까워질수록 물체의 이미지가 커져서 도심윤곽

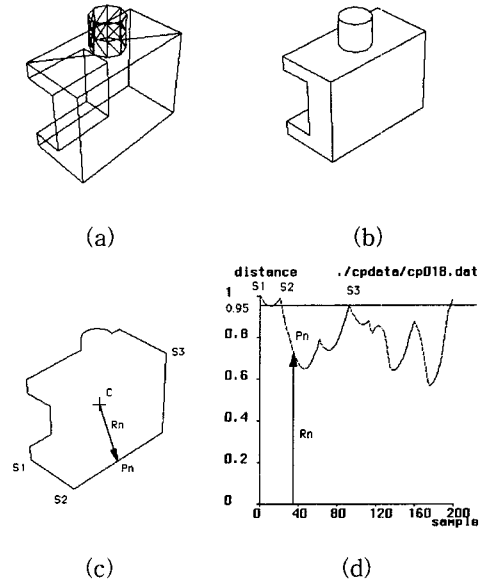


Fig. 4 The procedure to extract the centroidal profile feature from a CAD model: (a) Surfaces of the CAD model are approximated by polyhedrons. (b) Create a line drawing using the faceted hidden line removal. (c) Extract boundaries and a centroid from the line drawing. (d) A rotation invariant centroidal profile feature that are normalized with respect to scale is chosen to begin at point S1 that has the longest distance.

패턴 값이 커지게 된다. 따라서 도심윤곽패턴 값 중에서 가장 큰 값으로 모든 패턴 값들을 나눔으로써 이미지의 크기에 관계없는 도심윤곽패턴을 얻을 수 있다. (Fig. 4(d)) 이들 도심윤곽패턴들은 80개의 모델 자세 중 실제 자세와 가장 가까운 자세에 물체자세를 찾아내고 물체의 거리는 표준 작업 거리로 할당할 다음 반복적 자세 계산 방법에 의해서 정확한 자세와 거리가 계산되어 진다.

5. 대응관계의 저장과 검색

이차원 이미지의 형상특징과 삼차원 모델의 형상특징사이의 대응관계를 도심윤곽패턴을 이용하여 저장할 수 있는데 Fig. 5는 그 과정을 설명하고 있다. 먼저 학습단계에서는 삼차원 CAD 모델

을 임의의 모델관점에서 바라봤을 때 이차원 이미지를 얻게되고, 이 때 삼차원 모델 모서리(E1-E18)와 이차원 이미지 모서리(e1-e12) 사이의 대응관계가 얻어진다. 이차원 이미지의 윤곽선(B1-B8)을 구한 후 모델도심윤곽패턴을 구하는 과정에서 패턴의 각 데이터 점들이 어느 윤곽선에 속해있는지를 검토함으로써 각 데이터 점마다 대응하는 윤곽선 번호를 저장한다. 위에서 구한 대응관계들을 이용하여 결과적으로 모델도심윤곽패턴의 각 데이터 점들과 삼차원 모델의 모서리 사이의 대응관계를 알 수 있다.

실행단계에서는 카메라에 잡힌 입력 이미지로부터 모서리들을 추출한다(L1-L12). 물체를 배경으로부터 분리해낸 후 윤곽선(D1-D8)을 얻는다.

이 때 입력이미지의 모서리들과 윤곽선사이의 대응관계를 얻을 수 있다. 그 후 윤곽선으로부터 입력도심패턴을 추출함으로써 결과적으로 입력도심패턴의 각 데이터 점과 입력이미지의 모서리(D1-D8) 사이의 대응관계를 알 수 있다.

인공신경망에 의하여 입력도심윤곽패턴과 가장 비슷한 모델도심윤곽패턴이 찾아지면 두 패턴의

각 데이터 점 사이에는 일대일 대응관계가 성립한다. 따라서 입력도심윤곽 패턴의 데이터 점마다 삼차원 모서리의 대응관계가 얻어진다.

대응관계 검색의 목표는 입력이미지의 모서리(L1-L12)와 삼차원 모델의 모서리(E1-E18)사이의 대응관계를 얻는 것이다. 그러나 인공신경망을 이용하여 찾아낸 모델관점은 입력관점의 근사치에 불과하므로 같은 삼차원 모서리에 대응하는 모든 데이터 점들이 모두 입력이미지의 같은 모서리에 대응하는 것은 아니다. 그러므로 다음과 같은 투표방법을 사용한다. 입력도심윤곽패턴의 데이터 점들 중 동일한 입력 윤곽선에 추출된 데이터 점들에 대하여 이들과 대응하는 삼차원 모서리에 투표를 하도록 한 후 가장 많은 표를 얻은 삼차원 모서리를 이차원 윤곽선에 대응하는 삼차원 모서리로 삼는다.

6. 인공신경망의 학습과 물체인식

6.1 인공신경망의 학습

PRONET에서는 입력도심윤곽패턴을 모델도심윤곽패턴들 중에서 가장 비슷한 패턴으로 분류하기 위하여 3 계층 인공신경망이 사용되었다. Fig. 6에서 입력 계층은 N개의 노드로 구성되어 있는데 각 노드는 도심윤곽패턴의 데이터 점에 대응된다. 출력 계층은 M개의 노드를 가지고 있는데 각 노드는 모델 관점에 대응된다. 이 논문에서는 입력계층은 100개의 노드를 가지고 출력계층은 80개의 노드를 가진다.

학습 단계에서 인공신경망은 역전파 방법을 이용하여 80개의 모델 패턴들이 학습된다. Fig. 7은 인공신경망을 학습시키기 위해 입력 계층에 입력되는 전형적인 도심윤곽패턴과 그 때 출력계층에 할당되는 목표패턴을 보여준다. 목표패턴은 입력되는 도심윤곽패턴의 관점에 해당하는 노드에 0.9를 할당하고 나머지 노드에 -0.9를 할당함으로써 얻어진다.

인공신경망의 학습이 끝나고 나면 각 계층의 가중치들은 저장되었다가 물체를 인식하는 과정에서 다시 사용된다. 모델라이브러리에 여러 개의 CAD모델이 있는 경우에는 각 CAD 모델은 그 자신만의 인공신경망을 가지게 된다. 즉, 하나의 인공신경망은 하나의 CAD 모델에서 얻어진 도심윤곽패턴들에 의하여 학습된다.

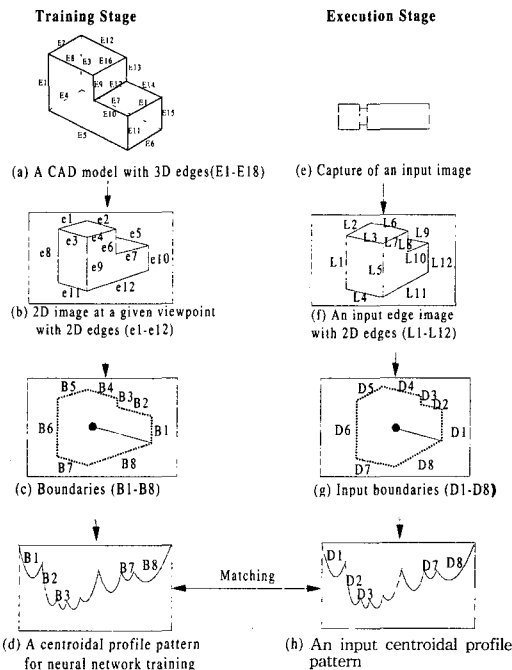


Fig. 5 Description of the procedure that assigns correspondences between the 2D edges of the real image and the 3D edges of the CAD model using single centroidal profile matching.

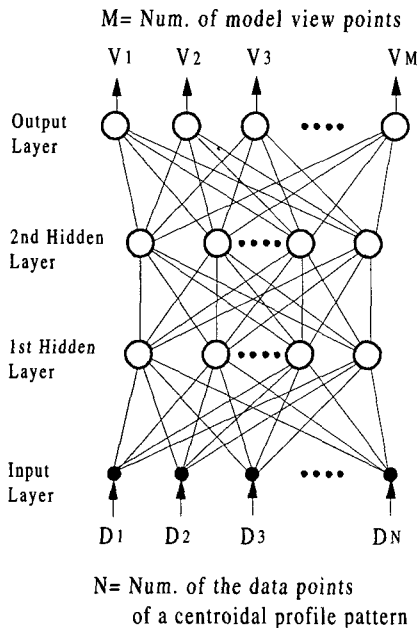


Fig. 6 The structure of a three-layer feed-forward neural network.

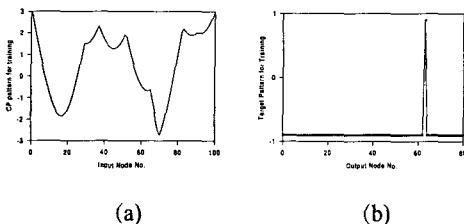


Fig. 7 Typical patterns for neural network training: (a) a model centroidal profile pattern (taken from the 65th view point) that is inputted to the input layer, and (b) the corresponding target pattern that is assigned to the output layer during the error back-propagation training.

6.2 인공신경망을 이용한 물체의 인식

실행단계에서는 먼저 로봇의 팔 끝에 장착된 카메라가 임의의 위치에서 임의의 자세로 놓여진 물체를 바라봄으로써 물체의 입력이미지를 얻는다. 이 입력이미지로부터 물체의 윤곽을 얻기 위하여 물체의 색상정보를 이용하여 물체와 배경을 분리한다. 즉, RGB(Red, Green, Blue) 형식의 입력 이미지를 Fig. 8(a), (b), (c)와 같이 HSI(Hue, Saturation, Intensity) 형식으로 전환한 다음 색상

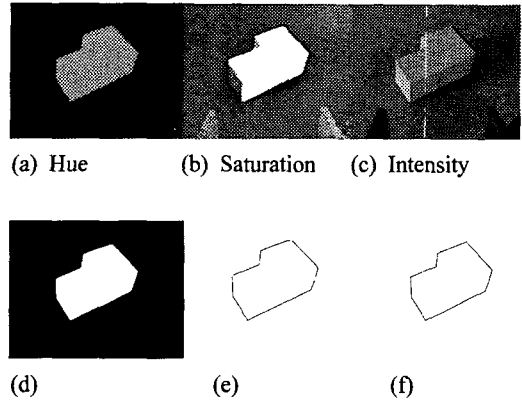


Fig. 8 Boundary extraction from the input image: (a),(b),(c) hue, saturation, and intensity images of the input image, (d) the image segmented from the hue image using thresholding, (e) the straight lines extracted from the segmented image, (f) the final boundary that does not have broken edges or missing corners. The centroidal profile pattern is extracted from this boundary.

(Hue)정보를 이용하여 Fig. 8(d)와 같이 물체를 배경으로부터 분리한다. 색상정보를 이용하여 물체를 분리하기 위해서는 물체가 단색이거나 배경이 단색이어야 한다는 가정이 전체되어야 한다.

분리된 물체형상으로부터 윤곽을 얻고 이 윤곽을 직선으로 근사시키고 끊어진 윤곽을 연결하여 Fig. 8(f)와 같이 잘 연결된 직선 윤곽을 얻는다. 이로부터 도심윤곽패턴을 얻는다.

실행단계에서는 물체의 이미지로부터 얻어진 입력도심윤곽패턴이 각 CAD 모델에 속한 인공신경망에 순차적으로 입력된다. 이 때 출력 노드 중에서 입력 도심윤곽패턴과 가장 비슷한 모델윤곽패턴을 얻을 수 있는 관점에 해당하는 노드가 가장 큰 값을 가진다. 각 인공신경망마다 가장 큰 값을 갖는 노드가 생길 수 있으며 이들을 올바른 답의 후보로 생각하여 정확한 자세계산과정을 통하여 올바른 관점을 찾아낸다.

가장 큰 노드가 정해지면 다음 세 가지 정보를 얻을 수 있다. 1) 인공신경망이 어느 CAD 모델에 속해있는가를 확인함으로써 물체의 종류를 알 수 있고 2) 출력 노드의 번호로부터 관점의 근사치를 알 수 있으며 3) 모델의 삼차원 모서리와 이미지의 이차원 모서리의 대응관계를 알 수 있다.

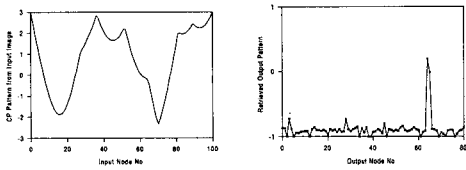


Fig. 9 (a) The centroidal profile pattern extracted from the input image: (b) Output from the neural network where the most corresponding viewpoints are the 64th and the 65th.

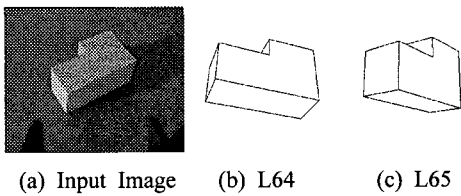


Fig. 10 (a) The input image, and (b), (c) the images taken from the viewpoints 64 and 65 respectively. L64 is the most similar to the actual viewpoint.

Fig. 9a는 입력 이미지로부터 얻어진 도심윤곽 패턴을 보여주고 Fig. 9b는 인공신경망의 출력 패턴을 보여주는데 64와 65번 노드가 큰 값을 나타내고 있다. Fig. 10은 입력이미지와 64번과 64번 관점에서 바라본 물체의 이미지를 보여준다.

물체의 자세는 관점자세각도(ϕ, θ, ψ)와 위치벡터(T_x, T_y, T_z)에 의해서 정해진다. 그 중에서 관점자세각도는 입력도심윤곽패턴과 가장 비슷한 모델도심윤곽패턴을 찾음으로써 그 근사치가 얻어진다. 관점자세각도들 중 ψ 각도는 이차원 이미지의 모니터 스크린 상의 회전을 나타낸다. 그런데 ψ 는 多觀點모델을 생성시킬 때 방향벡터(\vec{o})를 임의로 아래로 향하게 잡았으므로 실제 관점자세를 구하기 위해서는 그 값을 보정해 주어야 한다. 모델도심윤곽패턴을 구할 때 도심에서 패턴의 시작점으로 향하는 벡터를 구하여 모델윤곽패턴에 저장한 후, 입력도심윤곽패턴의 시작점과 모델도심윤곽패턴의 시작점을 비교함으로써 ψ 각도를 보정할 수 있다. Fig. 11은 ψ 각도 보정 전의 모델자세와 보정 후의 모델자세를 보여준다.

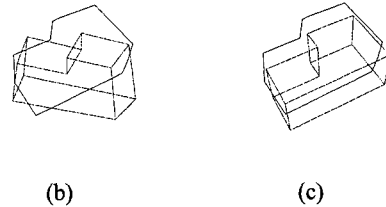
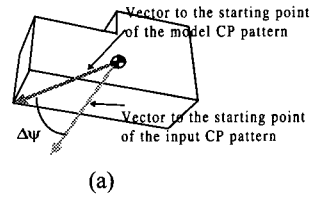


Fig. 11 (a) The ψ correction angle ($\Delta\psi$) (b) A model pose before the ψ correction (c) A model pose after the ψ correction.

위치벡터는 다음과 같은 가정을 함으로써 그 근사치를 얻을 수 있다. 물체가 이미지의 한가운데 있다고 가정하면 물체의 카메라에 대한 좌우상하 이동(T_x, T_y)는 없다고 놓을 수 있다. 카메라와 물체와의 거리(T_z)는 PRONET이 사용될 시스템의 작업거리로 정한다.

위에서 얻은 관점자세각도와 위치벡터의 근사치들을 실제값들과 차이가 있으나 반복적 자세계산 과정에 초기값들로 사용되어 실제값으로 빨리 수렴된다.

7. 정확한 자세의 계산

앞에서 구한 관점자세와 위치벡터의 근사치들로부터 정확한 물체의 자세를 구하기 위하여 Lowe⁽¹¹⁾와 Goldberg⁽¹²⁾가 제안한 반복적 모델자세 계산방법을 적용하였다. 이 방법은 먼저 카메라로부터 입력 이미지가 얻어지면, 물체의 모서리를 찾아 입력 모서리로 사용한다. 모델자세의 근사치를 이용하여 삼차원 모델을 입력 이미지 위에 투영시키고, 투영된 모델의 각 모서리에 대응하는 입력 모서리를 찾아낸다. 두 모서리 사이의 거리를 최소화하도록 뉴턴-랩슨 방법을 이용하여 반복적으로 물체자세를 바꾸어 가면 물체자세의 근사치는 정확한 값으로 수렴되어 간다. 부록은 그 자세한 내용을 설명하고 있다. 이 방법은 두 가지의 입력값을 요구한다. 첫째로 관점자세각도와 위치벡터로 표현되는 물체자세의 근사치가 필

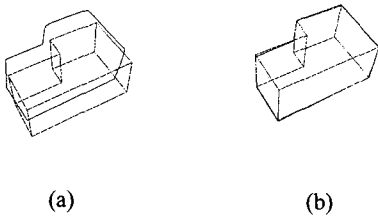


Fig. 12 Accurate pose calculation using the iterative model posing method: (a) the initial pose for iterative model posing after the ϕ correction (iteration 0), (b) the pose of the model after the first iteration.

요하다. 둘째로 삼차원 CAD 모델의 모서리와 카메라에서 얻어진 입력 모서리와의 대응관계가 필요하다.

PRONET 방법에서는 물체자세의 근사치와 입력 모서리와 모델 모서리 사이의 대응관계를 6.2 절에서 설명한 방법대로 인공신경망을 이용하여 찾아낸다. 이는 기존의 방법들이 모서리 사이의 대응관계를 찾아내기 위하여 가정과 확인 방법을 사용하는 것과 비교하면 인공신경망을 이용한 한번의 도심유평패턴의 매칭으로 대응관계와 자세의 근사치를 찾아낼 수 있으므로 계산시간의 단축을 가져왔다.

Fig. 12는 물체의 정확한 자세를 구하는 과정에서 물체의 초기위치와 첫 번 반복계산이 끝난 후의 정확한 물체의 위치를 보여준다.

인공신경망의 출력계층에서는 몇 개의 높은 값을 갖는 노드가 있을 수 있는데, 이들을 올바른 답의 후보로 간주한다. 이들 후보들에게 물체자세 계산과정을 수행시키면 올바른 답은 1-2번의 반복계산 만에 정확한 자세로 수렴이 되지만 그릇된 답은 발산하게된다. 이로써 여러 후보 중에서 올바른 답을 얻을 수 있다.

8. 실험 결과

PRONET은 Galileo 비디오 보드와 R4000 프로세서가 장착된 실리콘그래픽스 인디고 워크스테이션을 사용하여 3가지 형상의 물체에 대하여 물체인식 및 자세계산 실험을 수행하였다.

먼저 모델 라이브러리에 1개의 물체만 있을 경우에 카메라가 입력이미지를 잡는 때부터 물체를 인식하고 정확한 자세를 구하는 데까지 걸리는

Table 1 Time sharing of the total recognition time.

Stage	Contents	Time
Image processing	* Input image capturing * Color Segmentation * Edge detection * Boundary extraction * Input centroidal profile pattern extraction	16 sec.
Neural network matching	* Read weighting factors from files * Retrieve output from a neural network.	5 sec.
Accurate pose calculation	* Pose calculation	2-3 sec.

전체 인식시간은 물체의 형상에 따라 차이는 있지만 약 25초가 걸렸다. Table 1에서 보는 바와 같이 전체 인식시간 중에서 이미지 프로세싱에 16초, 인공신경망의 매칭에 5초, 물체의 반복적 자세계산을 포함한 그 외의 과정에 2-3초 걸렸다. 전체 물체 인식 시간은 이미지 프로세싱 과정에 전문 하드웨어를 사용하고 프로그램을 속도에 관하여 최적화 함으로써 인식시간을 더욱 줄일 수 있다.

PRONET의 모델 라이브러리에 하나 이상의 모델이 있다면 입력도심유평패턴이 각 모델에 속한 인공신경망에 순차적으로 입력되기 때문에 인식 시간은 물체의 수에 비례하여 증가하게 된다. 이 경우 이미지 프로세싱은 최초 1번만 수행하면 되고, 여기서 구해진 입력도심유평패턴은 각 모델의 인공신경망에 차례로 입력되기 때문에 물체가 모델라이브러리에 하나씩 추가 될 때마다 약 5초의 시간이 더 소요되었다. 이는 가정과 확인의 방법을 이용한 기존의 삼차원 물체 인식방법에서 모델 수의 증가가 계산 시간의 폭발적인 증가를 가져와 인식시간이 수분씩 걸리던 것을 크게 단축시킨 것이다.

물체 인식에 실패하는 경우는 이미지 프로세싱 과정에서 조명의 방향이나 밝기의 변화 등으로 인하여 물체의 윤곽이 제대로 얻어지지 않은 원인이 가장 많았다. 또 PRONET은 물체인식과 자세계산에서 윤곽특징만 사용하기 때문에 같은 윤곽을 가지지만 내부모양이 다를 경우에는 인식 및 자세계산에 실패할 경우가 있었다. 또한 구(球)처럼 모든 방향에서 윤곽특징이 동일한 경우에는 학습이 불가능하고 정육면체처럼 동일한 윤

특징이 여러 관점에서 얻어질 수 있는 대칭형상의 물체 경우에도 학습단계에서 인공신경망을 수렴시키기 어렵다.

9. 결 론

이 논문에서는 삼차원 물체를 인식하고 그 자세를 정확하게 구해주는 PRONET 시스템을 개발하였다. 삼차원 물체의 빠른 인식을 위하여 이차원 패턴 인식에 사용되는 인공신경망을 삼차원 물체의 인식문제에 적용시켰다. 또한 인공신경망과 반복적 자세계산방법을 연계하여 물체의 정확한 자세계산을 가능하게 하였다.

PRONET 방법은 다음과 같은 장점이 있다.

(1) 시간이 많이 소모되는 과정을 학습단계에서 미리 수행했기 때문에 실시간 실행 시에 물체를 빨리 인식할 수 있다.

(2) 모델의 삼차원 모서리와 이미지의 이차원 모서리 사이의 대응관계가 한 번의 도심윤곽패턴의 매칭으로 얻어져서 물체의 자세를 계산하는데 사용되므로 물체 인식시간을 줄였다.

(3) 전체 시스템이 CAD시스템과 잘 통합되어 새로운 모델을 쉽게 추가할 수 있다.

(4) PRONET 방법은 물체형상이 복잡해지더라도 그 영향은 학습단계에서 약간의 시간이 더 소요될 뿐 실행단계에서의 인식시간에는 영향을 미치지 않는다.

PRONET은 앞으로 부분적으로 가려진 물체의 인식, 같은 윤곽을 갖지만 내부가 다르게 생긴 물체의 인식, 등의 기능을 더욱 발전시킬 필요가 있다. 그러나 PRONET의 빠른 물체인식과 정확한 자세계산은 생산현장에서 여러 가지 작업을 자동화하는데 이용될 수 있을 것이다.

참고문헌

(1) 손영탁, 김재선, 조형석, 고광일, 1995, "로봇 아크용접 공정을 위한 용접접합부의 시각인식 및 추종," 대한기계학회논문집 제19권 제12호, pp. 3216~3229.

(2) Rajavelu, A., Musavi, M. T., and Shirvaikar, M. V., 1989, "A Neural Network Approach to Character Recognition," *Neural Networks*, Vol. 2, pp. 387~393.

(3) Gupta, L., Sayen, M. R., and Tammana, R., 1990, "A Neural Network Approach to Robust Shape Classification," *Pattern Recognition*, Vol. 23, No. 6, pp. 563~568.

(4) Nasrabadi, N. M. and Li, W., 1991, "Object Recognition by a Hopfield Neural Network," *IEEE Transaction on Systems, Man, And Cybernetics*, Vol. 21, No. 6, pp. 1523~1535.

(5) Bebis, G. N. and Papadourakis, G. M., 1992, "Object Recognition using Invariant Object Boundary Representations and Neural Network Models," *Pattern Recognition*, Vol. 25, No. 1, pp. 25~44.

(6) Brooks, T. D. and Kim, J. H., 1993, "An Approach to Three-Dimensional Object Recognition Using a Hybrid Hopfield Network," *IEEE SOUTHEASTCON '93*.

(7) Wang, J. Y. and Cohen, F. S., 1991, "3D Object Recognition and Shape Estimation from Image Contours using B-Splines, Unwarping Techniques, and Neural Network," *International Joint Conference on Neural Networks*, Singapore, Vol. 3, pp. 2318~2324.

(8) Poggio, T. and Edelman, S., 1990, "A Network That Learns to Recognize Three-dimensional Objects," *Nature*, Vol. 343, No. 18, January, pp. 263 ~266.

(9) Huttenlocher, D. P. and Ullman, S., 1990, "Recognizing Solid Object by Alignment with an Image," *International Journal of Computer Vision*, Vol. 5, No. 2, pp. 195~212.

(10) Ballard, D. H. and Brown, C. M., 1982, *Computer Vision*, Prentice-Hall, Englewood Cliffs, New Jersey.

(11) Lowe, D. G., 1987, "Three-Dimensional Object Recognition from Single Two-Dimensional Images," *Artificial Intelligence*, Vol. 31, pp. 355 ~395.

(12) Goldberg, R. R., 1993, "Pose Determination of Parameterized Object Models from a Monocular Image," *Image and Vision Computing*, Vol. 11, No. 1, Jan./Feb., pp. 49~62.

(13) Gonzalez, R. C. and Woods, R. E., 1992, *Digital Image Processing*, Addison Wesley.

부 록

반복적 자세 계산 방법

Fig. 13에서 카메라 좌표계로 표현된 점 p_c 는 지구 좌표계에 대하여 표현된 점 p_g 로 다음과 같이 변환이 된다.

$$p_g = {}^gR p_c + {}^gT \tag{8}$$

여기서 gR 은 3개의 자세각도로 표현될 수 있는 3 x 3 회전행렬이고 gT 는 3개의 이동변위로 표현되는 3 x 1 이동벡터이다. 즉 물체의 자세를 나타내는 자세인자는 6개이다. 역으로 p_c 는 식 (8)로부터 다음과 같이 얻을 수 있다.

$$p_c = (x, y, z)^T = {}^gR(p_g - {}^gT) \tag{9}$$

p_c 는 원근법에 의하여 카메라의 이미지 평면에 식 (9)과 같이 투영된다. 여기서 f 는 카메라의 초점거리이다.

$$p_i = (x^i, y^i)^T = (fx/z, fy/z)^T \tag{10}$$

x^i 와 y^i 를 6 개의 자세인자로 미분하는 것은 너무 복잡하고 서로 독립적이지 못하다. 따라서, Lowe 는 지구 좌표계로 나타내어진 자세인자들을 카메라 좌표계에 대한 자세인자로 다시 구성하였다. 즉 p_g 를 gR 로 회전시킨 후 카메라 좌표계에 대하여 D_x, D_y, D_z 만큼 평행이동을 시키는 것이다.

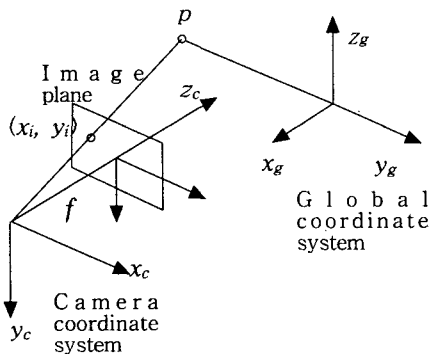


Fig. 13 Camera model.

이 때 식 (9)와 (10)은 다음과 같이 바뀐다.

$$p_c = (x, y, z)^T = {}^gR p_g \tag{11}$$

$$p_i = (x_i, y_i)^T = \left(\frac{fx}{z + D_z} + D_x, \frac{fy}{z + D_z} + D_y \right)^T = (fx c + D_x, fy c + D_y)^T \tag{12}$$

식 (12)에서 x_i 와 y_i 의 평행 이동량(D_x, D_y, D_z)에 대한 편미분은 쉽게 구할 수 있으나 x_i 와 y_i 의 회전각도에 대한 편미분은 여전히 구하기가 어렵다. 따라서 회전행렬(gR)의 초기치를 알면 그 값에서 카메라 좌표계의 x, y, z 축에 관한 물체의 자세각도의 미세 변화(ϕ_x, ϕ_y, ϕ_z)를 더해 감으로써 정확한 자세각도를 구할 수 있다. 식 (4)에서 얻어진 p_c 의 좌표 (x, y, z) 를 ϕ_x, ϕ_y, ϕ_z 로 편미분한 값은 선형화 되어서 Table 2와 같이 주어진다. 따라서 물체자세를 표시하는 인자를 P 라고 하면 $P = (\phi_x, \phi_y, \phi_z, D_x, D_y, D_z)^T$ 로 주어진다.

Table 2 Partial derivative of x, y, and z with respect to counter-clockwise rotation ϕ 's (in radian) about the coordinate axes.

	x	y	z
ϕ_x	0	-z	y
ϕ_y	z	0	-x
ϕ_z	-y	x	0

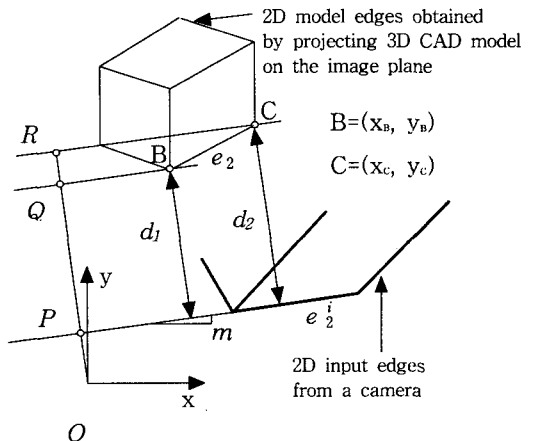


Fig. 14 Iterative model posing method using line-to-line correspondence.

Fig. 14에서 e_2 는 삼차원 CAD 모델을 모델 자체의 초기치를 이용하여 화면에 투영시킨 이미지의 모서리이고 e_2^i 는 e_2 에 대응하는 입력 이미지의 모서리를 나타낼 때 물체의 자세는 다음의 방법으로 정확한 자세로 수렴되어 간다.

e_2^i 의 직선식은 $y=mx+b$ 로 주어지고 e_2 의 끝점 B와 C에서 e_2^i 로의 법선거리는 다음과 같이 주어진다.

$$\begin{aligned} d_1 &= OQ - OP \\ &= (1/\sqrt{1+m^2})y_B - (m/\sqrt{1+m^2})x_B - b/\sqrt{1+m^2} \\ &= (y_B - mx_B - b)/\sqrt{1+m^2} \end{aligned} \quad (13)$$

$$\begin{aligned} d_2 &= OR - OP \\ &= (1/\sqrt{1+m^2})y_C - (m/\sqrt{1+m^2})x_C - b/\sqrt{1+m^2} \\ &= (y_C - mx_C - b)/\sqrt{1+m^2} \end{aligned} \quad (14)$$

법선거리 d 는 자세인자 P 의 함수이다. 따라서 d 를 최소화하려면 새로운 자세인자 $P + \Delta P$ 가 법선거리 d 를 0으로 만들게 하도록 ΔP 를 정하면 된다. 즉, 뉴턴-랩슨 방법을 이용하여 d 는 최소화 된다.

$$d(P + \Delta P) = d(P) + \frac{\partial d}{\partial P} \Delta P = 0 \quad (15)$$

$$\frac{\partial d}{\partial P} \Delta P = -d(P) \quad (16)$$

하나의 대응하는 모서리 쌍에 대하여 d_1 과 d_2 에 관한 2개의 식이 만들어진다. 즉 6개의 물체자세 인자들을 찾기 위해서는 적어도 3개의 대응하는 모서리 쌍을 찾아야한다. 식 (16)은 다음과 같이 표시된다.

$$\begin{aligned} \frac{\partial d_1^i}{\partial D_x} \Delta D_x + \frac{\partial d_1^i}{\partial D_y} \Delta D_y + \frac{\partial d_1^i}{\partial D_z} \Delta D_z + \\ \frac{\partial d_1^i}{\partial \phi_x} \Delta \phi_x + \frac{\partial d_1^i}{\partial \phi_y} \Delta \phi_y + \frac{\partial d_1^i}{\partial \phi_z} \Delta \phi_z = -\Delta d_1^i \end{aligned} \quad (17)$$

$$\begin{aligned} \frac{\partial d_2^i}{\partial D_x} \Delta D_x + \frac{\partial d_2^i}{\partial D_y} \Delta D_y + \frac{\partial d_2^i}{\partial D_z} \Delta D_z + \\ \frac{\partial d_2^i}{\partial \phi_x} \Delta \phi_x + \frac{\partial d_2^i}{\partial \phi_y} \Delta \phi_y + \frac{\partial d_2^i}{\partial \phi_z} \Delta \phi_z = -\Delta d_2^i \end{aligned} \quad (18)$$

Table 3는 법선거리를 각 자세인자들로 편미분한 값들을 나타내고 있다.

Table 3 Partial derivative of d_1^i and d_2^i with respect to each of the camera transform parameters where $a = 1/\sqrt{(1+m^2)}$, $c = 1/(z + D_z)$.

	d_1^i	d_2^i
D_x	$-ma$	$-ma$
D_y	a	a
D_z	$fc^2a\{mx_1 - y_1\}$	$fc^2a\{mx_2 - y_2\}$
ϕ_x	$fca\{mcx_1y_1 - z_1 - cy_1^2\}$	$fca\{mcx_2y_2 - z_2 - cy_2^2\}$
ϕ_y	$fca\{-m(z_1 + cx_1^2) + cx_1y_1\}$	$fca\{-m(z_2 + cx_2^2) + cx_2y_2\}$
ϕ_z	$fca\{my_1 + x_1\}$	$fca\{my_2 + x_2\}$

식 (17)과 (18)로부터 만들어진 선형방정식은 다음과 같이 표현된다.

$$[A][\Delta P] = [E] \quad (19)$$

여기서 n 이 대응하는 모서리 쌍의 갯수일 때, A 는 크기가 $2n \times 6$ 인 자코비안 행렬이고 ΔP 는 크기가 6×1 인 자세인자의 변화량 행렬이고, E 는 크기가 $2n \times 1$ 인 오차 행렬이다. n 이 3보다 클 경우에는 최소제곱법을 이용하여 다음과 같이 해를 구할 수 있다.

$$[A]^T[A][\Delta P] = [A]^T[E] \quad (20)$$

$$[\Delta P] = ([A]^T[A])^{-1}[A]^T[E] \quad (21)$$

여기서 구한 ΔP 를 이용하여 자세 인자를 새로 구하여 앞의 과정을 반복하여 계산함으로써 법선거리 d_1 과 d_2 를 최소화하는 자세인자를 구할 수 있다.