

<논 문>

## 스카라 로봇의 오프라인 프로그래밍을 위한 시각정보 보정기법

박창규\* · 손 권\*\*

(1995년 1월 27일 접수)

### A Visual Calibration Scheme for Off-Line Programming of SCARA Robots

Chang Kyoo Park and Kwon Son

**Key Words :** Off-Line Programming System(오프라인 프로그래밍 시스템), SCARA Robot(스카라 로봇), Position-Based Visual Servoing(위치기반 비주얼 서보잉), Predicted Image(예측이미지), Camera Image(카메라이미지), Perspective Projection(원근 투영)

#### Abstract

High flexibility and productivity using industrial robots are being achieved in manufacturing lines with off-line robot programmings. A good off-line programming system should have functions of robot modelling, trajectory planning, graphical teach-in, kinematic and dynamic simulations. Simulated results, however, can hardly be applied to on-line tasks until any calibration procedure is accompanied. This paper proposes a visual calibration scheme in order to provide a calibration tool for our own off-line programming system of SCARA robots. The suggested scheme is based on the position-based and image-based visual servoings, and the perspective projection. The scheme requires only one camera as it uses saved kinematic data for three-dimensional visual calibration. Predicted images are generated and then compared with camera images for updating positions and orientations of objects. The scheme is simple and effective enough to be used in real time robot programming.

#### 1. 서 론

오프라인 프로그래밍(off-line programming, OLP)은 산업용 로봇의 생산성 증대에 크게 기여하고 있다. 프로그램 상에서 변경된 로봇작업과 환경에 대하여 시뮬레이션을 수행하고, 그 결과를 생산라인 상의 로봇에 직접 적용함으로써, 시간과 비용을 절감할 수 있을 뿐만 아니라 예측하기 어려

운 문제나 위험요소도 사전에 제거할 수 있다. 컴퓨터 상에서 구현된 각종 작업에 대한 시뮬레이션 결과를 그대로 현장에 적용하기 위해서는 변화하는 작업정보가 수시로 OLP에 반영되어야만 한다. 따라서 OLP는 여러 시뮬레이션 기능 외에도 오프라인과 온라인의 상태를 비교·조정할 수 있는 보정기능을 갖추어야 한다.<sup>(1)</sup>

보정에서는 주로 시각정보를 이용하는데 이를 위한 비주얼 서보잉(visual servoing)은 크게 위치기반(position-based)과 이미지기반(image-based) 방식으로 분류된다.<sup>(2)</sup> 위치기반방식은 위치를 판단하

\*부산대학교 대학원 기계공학과

\*\*회원, 부산대학교 기계공학과 및 기계기술연구소

기 위하여 사진측량,<sup>(3)</sup> 스테레오비전,<sup>(4)</sup> 레인지이미지<sup>(6)</sup> 등을 이용한다. 이미지기반방식은 이미지의 특징을 바탕으로 로봇위치를 이동시키는 방법으로 기준이미지와 측정된 이미지의 비교로 로봇위치를 제어한다.<sup>(6)</sup>

본 연구에서는 PC용 오프라인 프로그래밍 시스템(POLPS)<sup>(7)</sup>에서 로봇의 위치를 보정하는 방법을 제안하였다. 위치기반과 이미지기반의 두 방식을 접합하여 사용하고, 원근투영법을 사용하여 OLP 내에서 실제 스카라 로봇의 위치를 파악할 수 있는 방법을 다루었다. 프로그램을 통해서 예측이미지를 계산하고, 실험에 의해 카메라이미지를 얻은 후, 물체의 특징점을 바탕으로 두 이미지를 비교하였다. 카메라를 이용할 때 제안된 방법으로 카메라 높기와 이미지에 나타난 물체길이 사이에 성립하는 간단한 관계식을 얻었다. 카메라에 의한 정보획득과 그 처리가 POLPS 상에서 행해지도록 소프트웨어를 구성하였으며, 실험을 통해 예측이미지와 카메라이미지를 이용하여 로봇의 보정이 가능함을 확인하였다. 개발된 보정법은 원근투영법을 사용하기 때문에 직접적인 3차원 정보의 계산이 요구되지 않아 실시간처리에 도움을 줄 수 있다. 따라서 시각정보에 의해 보정된 시뮬레이션 결과를 생산라인에 적용할 수 있는 본 연구의 방법은 로봇보정, 궤적추종, 컴플라이언스 운동시 정확한 작업을 수행하는데 도움을 줄 수 있으며, 조립라인에서 공작물의 위치를 감지하는 데에도 사용될 수 있다.

## 2. 이미지의 생성

카메라이미지를 이용하여 로봇의 위치를 검색하고, 그 위치와 자세를 보정해 주기 위해서는 로봇 특성을 고려한 시각정보 처리기법이 요구된다. 스카라 로봇을 위한 POLPS에서 시각정보를 이용한 보정방법은 위치기반 방식을 기본으로 하나, 두 이미지의 특징을 비교하는 이미지기반 방식도 병행하고 있다. 이는 예측된 로봇의 위치에서 알려진 3차원 CAD 정보에 의해 만들어진 예측이미지와 현재의 위치에서 추출되는 카메라이미지를 2차원 상에서 비교하는 방법으로 그 적용단계는 Fig. 1과 같다.

카메라이미지와 비교되는 예측이미지를 생성하기 위해서 프로그램 상의 CAD 정보로부터 2차원 이미지를 생성한다.<sup>(8)</sup> 본 연구에서 모든 대상물체는

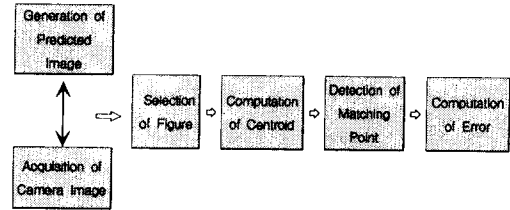


Fig. 1 Process of visual calibration scheme

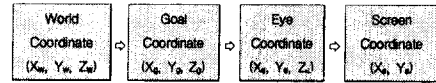


Fig. 2 Coordinate transformations

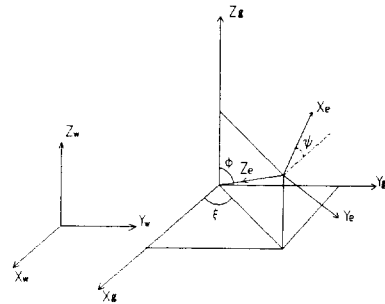


Fig. 3 Coordinate systems

곡선과 곡면이 없는 직선, 꼭지점, 평면으로 구성된 물체라고 가정한다. 카메라에서의 투영은 렌즈 축방향의 수직거리에 따르는 원근법을 사용할 수 있다고 가정한다. 두 이미지의 비교시 경계추출에 의해 생성된 경계선을 이용하며, 모든 경계선은 연결되어 있고 단일두께를 가진다고 가정한다. 카메라이미지로부터 물체의 내부선을 고려하지 않는 단순 윤곽을 추출하며, 프로그램 상에서 이것과 비교하기 위해 단순 윤곽선을 가지는 예측이미지를 생성한다.

### 2.1 스크린좌표계의 변환과 투영

Fig. 2는 예측이미지를 구성하기 위한 월드좌표( $X_w, Y_w, Z_w$ )에서 스크린좌표( $X_s, Y_s$ )로의 변환과정을 나타내며, 각 좌표계 사이의 관계는 Fig. 3과 같다. 월드좌표에서 목표좌표( $X_g, Y_g, Z_g$ )로의 변환은 월드좌표계의 원점에서 목표좌표계의 원점까지의 병진변환에 의해서 이루어진다. 목표좌표에서 시각좌표( $X_e, Y_e, Z_e$ )로의 변환행렬은 다음 식과 같다.

$$T_g = \begin{bmatrix} -c\phi c\xi s\psi - c\psi s\xi c\psi c\xi - c\phi s\psi s\xi & s\psi s\phi & 0 \\ c\psi c\phi c\xi - s\psi s\xi & c\xi s\psi + c\psi c\phi s\xi & -c\psi s\phi & 0 \\ -c\xi s\phi & -s\phi s\xi & -c\phi & \rho \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

여기서,  $c$ 는  $\cos$ ,  $s$ 는  $\sin$ ,  $\phi$ 는  $X_g$ 축과  $X_e$ 축의 교각, 그리고  $\rho$ 는 시각좌표계 원점에서 목표좌표계 원점까지의 거리를 나타낸다. 시각좌표에서 스크린좌표 ( $X_s, Y_s$ )로의 변환은 시각좌표계의  $Z_e$ 축 거리에 대한 원근투영에 의하여 행해진다.

2.2 시야 내·외부 판별

물체가 시야 내·외부에 있는 지를 판별하는 것은 이미지의 비교에서 매우 중요하다. 왜냐하면 물체가 시야영역 안에 있다고 판단되는 경우에만 예측이미지를 화면에 나타내기 때문이다. 물체의 기본요소인 점, 선, 면 등에 대한 시야 내·외부 판별법은 다음과 같다.

예측이미지 상의 점에 대해서는 Fig. 4의 시각좌표계 원점에서 점 ( $x, y, z$ )를 잇는 선과  $z$ 축이 이루는 각도가  $\theta_e$ 보다 작으면 시야영역 안이라고 판정한다.  $\theta_e$ 는 시야영역 범위를 나타내는 각도로서 시각좌표계 원점과 취득된 영상 사각형의 내접원의 원주각이 된다. 선이  $Z_e$ 축과 이루는 각도  $\theta$ 는 다음과 같이 표시된다.

$$\theta = \tan^{-1}\left(\frac{\sqrt{x^2 + y^2}}{z}\right) \quad (2)$$

예측이미지 상의 선에 대한 시야 내·외부 판별을 위해서 다음과 같이 선을 분류한다.

경우 1, 선분을 이루는 두 끝점이 모두 시야 안에 있는 경우.

경우 2, 한 끝점은 시야 안, 다른 끝점은 시야 밖에 있는 경우.

경우 3, 두 끝점은 시야 밖에 있지만 선분중 일부가 시야 안에 있는 경우.

경우 4, 선분의 전부가 시야 밖에 있는 경우.

Fig. 5는 분류된 선을 나타내며, 그림에서 원은 시야경계를 나타낸다. 경우 1은 선분 전체를 도시하며, 경우 2와 3은 선분중 시야에 포함된 부분을 구하여 포함된 부분만 나타내며, 경우 4는 화면에 전혀 나타나지 않는다. 경우 3과 4의 구별은 시야 영역을 이루는 원주의 중심선에서 최소거리를 갖는 검색대상 선분 상의 점이 시야 안에 존재하면 선분중 일부가 시야 안에 있다는 성질을 이용한다.

예측이미지 상의 면에 대해서도 다음과 같이 분류한다.

경우 1, 면이 전부 시야 안에 있는 경우.

경우 2, 면의 꼭지점 중 일부만 시야 안에 있는

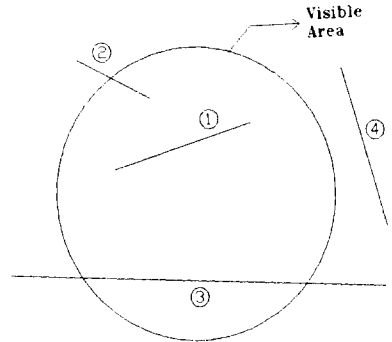


Fig. 5 Classification of lines

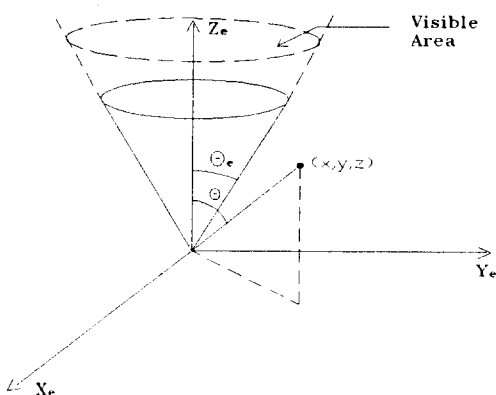


Fig. 4 Visible area

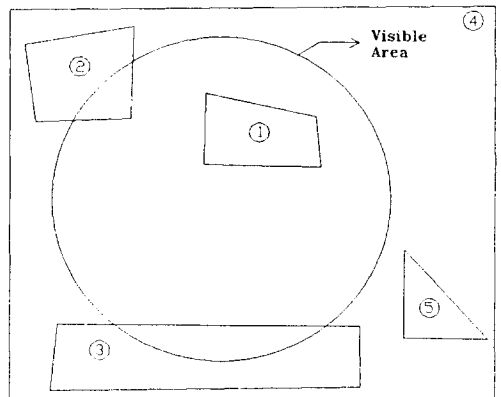


Fig. 6 Classification of faces

경우,

경우 3, 면의 꼭지점은 모두 시야 밖에 있지만 모서리 중 일부가 시야 안에 있는 경우,

경우 4, 면의 꼭지점이 모두 시야 밖에 있지만 면이 시야영역을 완전히 감싼 경우,

경우 5, 면의 일부가 시야 밖에 있는 경우.

Fig. 6에 위와 같이 분류된 면을 나타내었으며, 여기서도 원의 내부는 시야를 나타낸다. 경우 1은 면 전체를 도시하고, 경우 2, 3, 4는 면의 일부만 화면에 도시하며, 경우 5는 면을 전혀 도시하지 않는다. 경우 3의 처리를 위해서 면을 이루는 모든 모서리에 대해 선을 판별하는 방식으로 선분의 시야검사를 수행하여, 모서리중에서 시야 안에 포함되는 것이 있는 지를 검색한다. 경우 4와 5를 구분하기 위해서 면과 시야영역 원후의 중심선과의 교점을 검사하여, 그 교점이 면 안에 있으면 면의 일부가 시야영역 안에 있다고 판단한다. 이때 교점의 내·외부 판별은 2차원 평면에 투영한 면의 꼭지점과 교점이 이루는 내각의 합이  $360^\circ$ 이면 교점이 면의 내부에 있다는 성질을 이용한다.

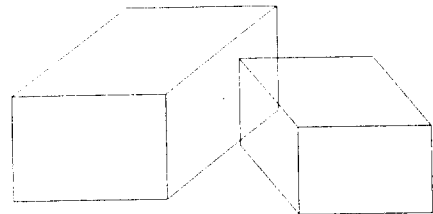
2.3 기준 윤곽선도형의 생성

카메라에서 얻어진 이치화(thresholding)<sup>(9),(10)</sup>된 이미지와 비교할 이미지를 프로그램 상에서 제시하기 위하여 단순 윤곽선도형(silhouette figure)을 생성한다. 단순 윤곽선도형을 생성하기 위해 깊이 검색법(depth sort method)<sup>(11)</sup>을 응용하여 적용하였다. Fig. 7는 기준 윤곽선도형을 구하는 단계를 나타낸다.

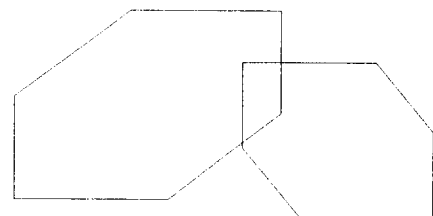
윤곽선도형을 얻기 위해서는 먼저 물체의 3차원 정보를 저장하여 계산에 이용할 수 있는 솔리드 모델의 자료를 생성해야 한다. 본 연구에서는 자료의 추가와 삭제가 용이한 고리형(linked list)의 자료구조<sup>(12)</sup>를 채택하였다. 와이어프레임 모델에서 은선과 은면을 제거하기 전에 시점에서 보이지 않는 배면을 미리 소거하면 계산시간을 줄일 수 있다. 면의 법선벡터와 물체의 시선벡터 사이의 각도가  $90^\circ$ 보다 작으면 그 면은 시점으로부터 보이고,  $90^\circ$ 보다 크면 보이지 않으므로 이 배면을 계산에서 제

외시킨다. 배면이 제거된 물체에서 내부선을 삭제함으로써 단순 외곽선도형을 생성할 수 있다. 선분이 배면을 이루는 모서리가 아니면 이 선분은 물체의 내부선을 이루는 모서리이므로 제거한다.

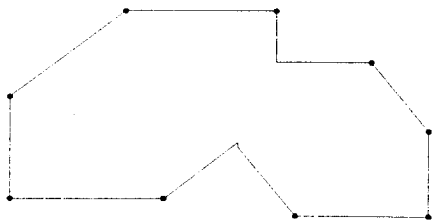
생성된 단순 외곽도형끼리 교차할 때에는 합쳐진 단순 윤곽선도형을 생성한다. 먼저 스크린좌표에서 좌표축을 기준으로 가까운 외곽도형의 가장 먼 꼭지점과 먼 외곽도형의 가장 가까운 꼭지점의 좌표값을 비교하여 교차여부를 판단한다. 교차가 가능하다고 판단되는 경우에는 외곽도형의 모든 모서리를 비교하여 어느 모서리에서 교차가 발생하였는지 알아낸다. 교차가 일어나지 않는다고 판단되는 경우에는 한 도형이 다른 도형에 완전히 포함되는 지를 검사한다. 포함검사는 면의 한 꼭지점이 다른 면의 꼭지점들과 이루는 각도의 합이  $360^\circ$ 이면 그면이 다른 면의 내부에 있다는 성질을 이용한다. Fig. 8의 (a), (b), (c)에 각 단계별 결과를 나타내었다.



(a) Figures with rear faces eliminated



(b) Exterior figures



(c) Silhouette figure

Fig. 8 Generation of predicted image

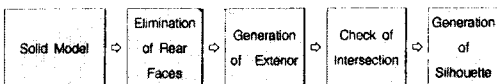
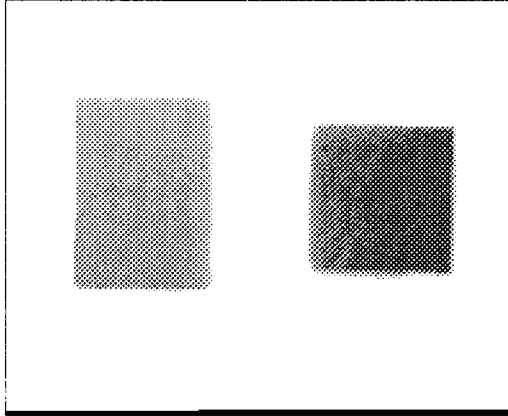
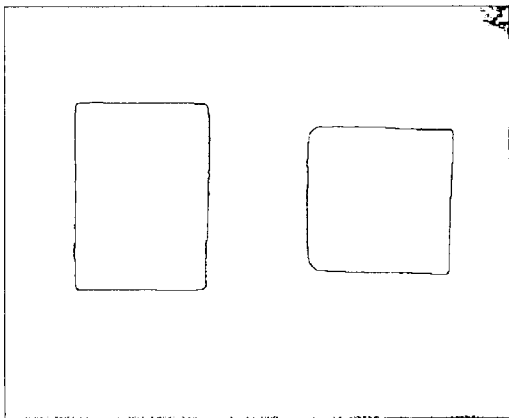


Fig. 7 Generation of silhouette figure



(a) Gray scale images



(b) Laplacian edges

Fig. 9 Generation of camera image

#### 2.4 입력이미지로부터 경계선 및 특징점 추출

카메라에서 얻어진 초기의 그레이스케일(gray scale) 이미지에서 경계선이 추출된 이미지를 얻기 위하여 다음의 두 단계를 수행한다. Fig. 9(a)에 나타난 그레이스케일 이미지에서 이진화에 의해 배경과 물체가 흑백으로만 구별되어지는 이진 이미지를 얻어 연속적인 경계선을 추출한 후, 라플라시안(Laplacian) 방법<sup>9)</sup>을 추가로 사용함으로써 OLP를 통해 생성되는 예측이미지와 비교할 수 있는 Fig. 9(b)와 같은 단일 두께의 경계선을 갖는 이미지를 구한다. 이 때 경계선의 두께가 2개 이상의 픽셀이 되는 경우에는 보다 안쪽에 위치한 픽셀을 경계선상의 점으로 간주하였다. 이와 같이 단일 두께의 연속적인 경계선을 갖는 단순 윤곽선도형을 추출하여 카메라이미지를 생성하였다.

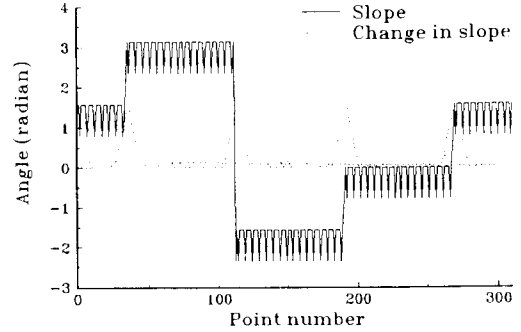


Fig. 10 Corner detection

카메라이미지에서 특징점을 추출하기 위한 방법은 다음과 같다. 두 이미지에서 2차원 오차를 계산하기 위해서는 카메라 시야 내에서 가장 잘 보이는 도형을 선정하여 꼭지점을 찾는다. 잘 보이는 검색대상 도형을 간단히 선정하기 위해 둘레의 길이가 제일 큰 도형을 선택하고, 그 꼭지점을 특징점으로 삼는다. 본 연구에서는 꼭지점을 찾기 위하여 모서리를 따르는 1차 미분값을 계산하여 미분값의 변화가 큰 점을 찾았다. 점 주위에서 한칸 간격으로 기울기를 검색하면 픽셀 단위의 이미지에서 기울기의 변화가 심하여 꼭지점을 제대로 찾아내기 어렵다. 따라서 검색대상 도형에서 한점 주위의 양쪽방향으로 여러칸 건너 떨어진 점에 대한 1차 미분값을 계산하여, 그 차가 국부적으로 최대인 점을 꼭지점으로 선정하였다. 이와 같은 방법으로 추출된 경계선의 기울기를 계산하면 필터링효과도 얻을 수 있다. Fig. 10은 사각형모양의 도형에 대해 한점 주위에서 한칸씩 떨어진 점과 다섯칸씩 떨어진 점의 기울기를 계산한 결과를 나타낸다. 그림에서 실선은 한칸 간격의 벡터 기울기값을 표시하고, 점선은 다섯칸 간격의 기울기의 변화값을 표시한다. 점선의 결과를 이용하는 것이 실선의 경우에 비해 정확하게 꼭지점에서의 각도변화를 판단할 수 있다. 그림에서 한칸 간격을 나타내는 실선에서 기울기가 급격히 변하는 곳이 다섯칸 간격의 결과에서 국부 최대점에 해당된다는 것을 확인할 수 있으며, 이 때의 값은  $90^\circ$ 부근이 된다.

### 3. 로봇의 위치오차

2차원 평면 상에서 예측이미지와 카메라이미지를 비교하여 3차원 좌표 상의 오차를 계산한다. 3차원

오차를 산출하기 위해서는 두 이미지에서 서로 대응되는 점을 찾아야 한다. 스카라 로봇이 수직 아래 방향을 향한다는 특성을 이용하면 두 개의 대응점만으로 오차를 모두 계산할 수 있다. 대응점을 찾기 위하여 예측이미지에서 선정된 검색대상 도형의 도심을 구한다.<sup>(13)</sup> 카메라이미지에서 도심을 중심으로 검색하여 근접점에서 대응점을 찾는다. 예측이미지에서 얻어진 도심을 이용하여 카메라이미지의 대응점을 찾음으로써 대응점 검색의 어려움을 해결한다.

본 연구에서는 원근투영법<sup>(13)</sup>을 사용하여 두 이미지의 대응점에서 로봇의 위치오차를 간단히 계산할 수 있는 방법을 제시한다. 오차는 Fig. 4의 Z축을 중심으로 하는 회전오차, Z축 방향의 높이오차, X와 Y축 방향의 병진오차로 나뉘어 계산된다.

3.1 원근투영법

원근투영법을 사용하면 카메라높이 Z와 영상 내의 선분길이 l사이의 관계식은 다음과 같다(Fig. 11 참조).

$$L \cdot D = l \cdot (Z + z_c) \tag{3}$$

여기서, L은 l에 대응되는 실제 길이, D는 카메라 초점거리, 그리고 z<sub>c</sub>는 카메라 끝에서 렌즈중심까지의 거리이다.

물체에서 카메라 끝까지의 거리인 카메라높이 Z에 대해 식 (3)을 나타내면 다음과 같다.

$$Z = \frac{L \cdot D}{l} - z_c \tag{4}$$

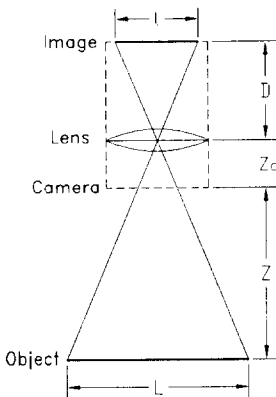


Fig. 11 Perspective projection

식 (4)에서 L은 알려져 있으므로 실험을 통해서 D와 z<sub>c</sub>를 얻을 수 있다면, 이 식을 이용하여 카메라 높이 Z에서 영상 내의 길이 l을 구하거나, 역으로 l에서 Z를 간단히 계산할 수 있다. 즉 Z값을 바꾸면서 시선방향에 수직한 평면에 놓여진 길이를 측정 한 후, 그 측정된 높이와 길이에 최소제곱법<sup>(14)</sup>을 적용하면 미지수인 D와 z<sub>c</sub>를 구할 수 있다. 최소제곱법을 이용하여 D와 z<sub>c</sub>를 구하는 식은 다음과 같다.

$$D = \frac{n \sum_{i=1}^n \frac{L}{l_i} f(Z_i) - \sum_{i=1}^n \frac{L}{l_i} \cdot \sum_{i=1}^n f(Z_i)}{\sum_{i=1}^n \frac{L^2}{l_i^2} - \left( \sum_{i=1}^n \frac{L}{l_i} \right)^2} \tag{5}$$

$$z_c = \frac{-\sum_{i=1}^n (Z_i) + \sum_{i=1}^n \frac{L}{l_i} D}{n} \tag{6}$$

여기서, n은 측정횟수, l<sub>i</sub>와 Z<sub>i</sub>는 i번째 측정값, 그리고 f(Z<sub>i</sub>)는 측정높이 Z<sub>i</sub>(i=1, 2, ..., n)를 계산한 값이다.

원근투영법을 사용함으로써 카메라높이와 영상 내의 길이 사이의 관계를 식 (4)와 같이 간단화시킬 수 있다. 이 방법은 기존의 연구에서 실제 물체 크기와 이미지 크기의 관계식에서 사용되는 스케일 인자를 높이에 따라 별도로 구하지 않고서 높이와 길이 사이의 관계에서 얻을 수 있다는 장점이 있다.

3.2 회전오차

Fig. 12에서와 같이 예측이미지에서 수평선과 모서리 P<sub>1</sub>P<sub>2</sub>가 이루는 각도가 η이고, 카메라이미지에서 모서리 C<sub>1</sub>C<sub>2</sub>가 이루는 각도가 η<sub>2</sub>일 때, η과 η<sub>2</sub>의 차이가 3차원 좌표상에서 회전오차 E<sub>η</sub>가 된

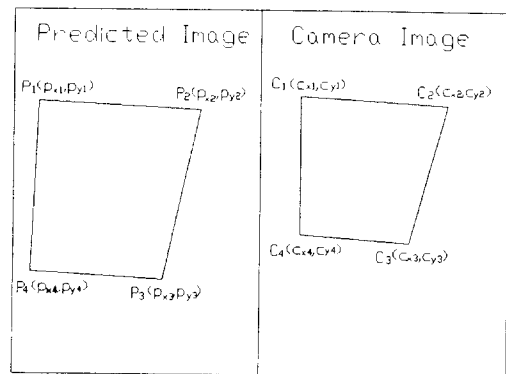


Fig. 12 Comparison of images

다. 따라서  $E_y$ 를 구하는 식은 다음과 같다.

$$E_y = \eta_2 - \eta_1 = \tan^{-1}\left(\frac{C_{y2} - C_{y1}}{C_{x2} - C_{x1}}\right) - \tan^{-1}\left(\frac{p_{y2} - p_{y1}}{p_{x2} - p_{x1}}\right) \quad (7)$$

### 3.3 높이오차

카메라이미지와 예측이미지를 비교하여 계산되는 높이오차  $E_z$ 를 다음과 같이 정의한다.

$$E_z = Z_2 - Z_1 \quad (8)$$

여기서,  $Z_1$ 은 OLPS 내에 입력된 카메라의 높이이며,  $Z_2$ 는 렌즈중심에서 물체까지의 최단 수직거리로 Fig. 11에서의  $(z_c + Z)$ 에 해당된다.

Fig. 13은 원근투영법을 이용하여 높이오차의 계산에 필요한  $Z_2$ 를 구하는 원리를 나타낸다. 그림에서  $L$ 은 실제 길이  $L^*$ 에 대해 카메라축에 수직한 평면에서 상으로 나타날 수 있는 길이이다.  $Z_2$ 는 영상 내의 길이  $l$ 에 대한  $L$ 의 비에 비례하며, 고려하는 물체의 선분이 카메라축과 수직을 이룬다면 다음과 같이 간단히 구해진다.

$$Z_2 = D \frac{L}{l} \quad (9)$$

물체의 선분이 카메라축과 수직이 아닌 경우에는 Fig. 13에 나타난 바와 같이  $L^*$ 에 대해 수직평면 상에서 대응되는 길이  $L$ 을 구한 후에 식 (9)를 적용한다. 이 때  $L$ 을 구하는 식은 다음과 같다.

$$L = L^* \frac{\sin(\alpha)}{\sin(\beta)}$$

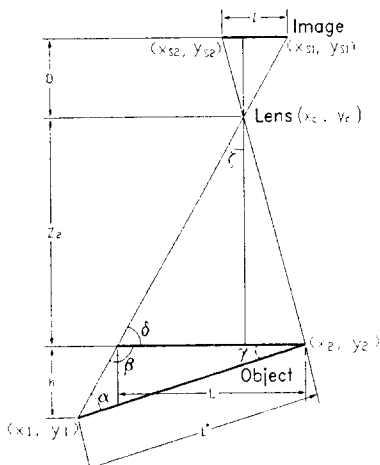


Fig. 13 Computation of height error

여기서,

$$\begin{aligned} \alpha &= \pi - \beta - \gamma \\ \beta &= \pi - \delta \\ \gamma &= \tan^{-1} \frac{h}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \\ \delta &= \pi/2 - \zeta \\ &= \pi/2 - \tan^{-1} \frac{\sqrt{(x_{s1} - x_c)^2 + (y_{s1} - y_c)^2}}{D} \end{aligned}$$

위 식에서  $\gamma$ 는 선분  $L^*$ 가 수평면과 이루는 각,  $h$ 는 선분 양끝점의 높이 차이,  $(x_1, y_1)$ 과  $(x_2, y_2)$ 는 물체의 선분을 나타내는 좌표로 OLPS의 3차원 CAD 정보로부터 얻어지는 값,  $(x_c, y_c)$ 는 렌즈중심의 위치로 화면의 중앙에 해당하는 좌표값, 그리고  $(x_{s1}, y_{s1})$ 과  $(x_{s2}, y_{s2})$ 는 각각 이미지 상에서 선분 양끝점의 좌표값이다.

### 3.4 병진오차

병진오차는 다음과 같은 순서로 계산된다. 앞에서 계산된 회전오차와 높이오차 만큼 단말효과기의 위치를 보정하여 예측이미지를 다시 구하고, 새로운 예측이미지와 카메라이미지에서 대응되는 위치를 비교하여 수평오차  $E_x$ 와 수직오차  $E_y$ 를 계산한다. 이 때 이미지상에서의  $x, y$ 축과 월드좌표계의  $X, Y$ 축이 서로 일치하지 않으므로, 이미지 상에서 계산된 병진오차를 단말효과기의 회전각도를 고려하여 월드좌표계에서의 병진오차로 변환시킨다. 이와 같이 월드좌표계에서 구해지는 수평과 수직 방향의 병진오차는 다음과 같이 표시된다.

$$E_x = \frac{Z_2 \cdot \lambda_x}{D} \quad (10)$$

$$E_y = \frac{Z_2 \cdot \lambda_y}{D} \quad (11)$$

여기서,  $\lambda_x$ 와  $\lambda_y$ 는 월드좌표계의  $X$ 와  $Y$ 축 방향으로 계산된 이미지 상에서의 병진오차를 나타내는 것으로 Fig. 11의  $l$ 과 관련된 오차이며,  $\lambda_x$ 와  $\lambda_y$ 는 이미지 평면에 수직한 축에 대한 회전변환에 의해 다음과 같이 구해진다.

$$\begin{bmatrix} \lambda_x \\ \lambda_y \end{bmatrix} = \begin{bmatrix} \cos \epsilon & \sin \epsilon \\ -\sin \epsilon & \cos \epsilon \end{bmatrix} \begin{bmatrix} p_{nx} - c_x \\ p_{ny} - c_y \end{bmatrix} \quad (12)$$

여기서,  $\epsilon$ 은 수평면 상에서 월드좌표계와 단말효과기 좌표계가 이루는 각도이며,  $c_x$ 와  $c_y$ 는 Fig. 12의 카메라이미지에서 선정된 한 꼭지점의 좌표이

며,  $D_{mx}$ 와  $D_{my}$ 는 회전오차와 높이오차를 없앤 새로운 예측이미지에서 대응점의 좌표이다.

### 4. 실험

본 논문에서 제안한 보정방법의 타당성을 검증하기 위해서 실험을 수행하였다. 제안된 오차계산법이 원근투영법에 근거하고 있으므로, 본 실험에서는 식 (4)의 적용 타당성을 증명하는 데 중점을 두었다. 실험에 사용된 시각정보처리 시스템은 Fig. 14와 같이 카메라, 영상처리 보드, 비디오 모니터, 개인용 컴퓨터 등으로 이루어졌다.

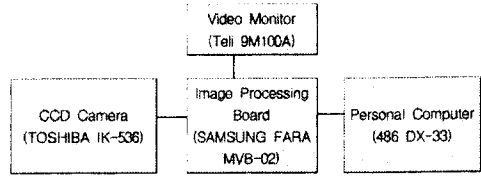


Fig. 14 Vision system

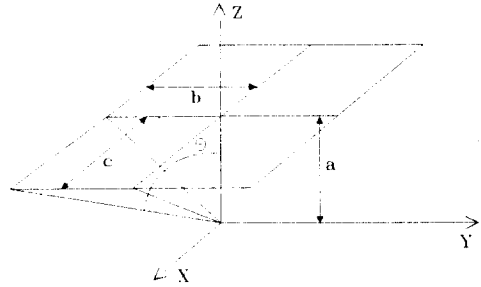


Fig. 15 Measurement of visible area

#### 4.1 카메라 사양의 측정

카메라 입력각의 정보는 시야영역을 측정하기 위해 반드시 필요하다. 입력각의 계산은 일정거리에서 시선 방향에 수직한 면에서 시야영역 내에 들어오는 길이를 이용하여 행해진다. 이를 위해 먼저 카메라의 위치를 정한 후, 카메라의 방향이 수직 아래방향을 향하도록 하였다. 카메라의 위치를 정할 때 카메라 렌즈의 4분원 지점에 주를 설치하고, 이를 이용하여 실제 카메라의 원점과 월드좌표계의 원점이 일치하게 하였다. 카메라의 수평위치를 월드좌표계 원점에 향하게 하고, 이미지좌표계와 월드좌표계의 원점을 일치시켜 그 수직성을 확보하였다.

카메라에서 입력받을 수 있는 시야각도  $\theta_c$ 는  $\theta$ 의 두배 값을 가진다(Fig. 15 참조). 주어진 카메라에서  $\theta$ 는 14.16°인데, 이는 그림에서 a, b, c의 합수로 간단히 표시될 수 있어 쉽게 계산된다. 실험에서 a, b, c는 각각 360, 65, 49 mm로 측정되었다.

초점거리의 계산에 사용되는 식 (4)에서 D와  $z_c$ 를 구하기 위한 측정은 두가지의 경우에 대해 행해졌다. 첫번째는 카메라 높이 변화폭이 작은 경우로 카메라 높이를 250에서 350 mm의 범위에서 변화시켰으며 측정간격을 5 mm로 잡았다. 두번째는 250에서 600 mm 까지의 높이에 대해 10 mm의 간격으

로 측정하였다. 두 경우에 대해 D와  $z_c$ 를 계산한 결과를 Table 1에 기재하였다. 동일한 카메라에 대해 두 경우에 상이한 D와  $z_c$ 값을 얻을 수 있다.

Table 1에서 평균오차는 동일한 카메라높이에서 측정된 이미지 상의 길이와 식 (4)를 이용하여 계산된 길이의 차이의 평균을 나타낸다. 높이 변화가 100 mm로 작은 경우에는 원근투영법을 사용한 결과가 측정치와 잘 일치하여, 그 차이의 평균이 0.5 픽셀 이하였다. 높이 변화폭이 350 mm로 큰 경우에는 계산 결과가 측정치와 평균 2.8 픽셀 정도로 큰 차이를 보였다. 오프라인 프로그램 POLPS의 대상인 스카라 로봇의 최대 높이가 357 mm이므로 대부분의 작동높이에서 원근투영법을 사용하여도 1 픽셀 미만의 정도를 유지할 수 있다. 1 픽셀 미만의 오차는 주어진 모니터의 분해능을 가지고는 모니터 상에서 나타낼 수 없기 때문에 원근투영법에 의해 계산된 결과는 충분히 정확하다고 간주된다.

Fig. 16의 (a)와 (b)는 위의 두 경우에 대해 측정구간 전역에 걸친 범위에서 측정과 계산결과를 나타내는 그래프이다. 카메라높이가 250~350 mm

Table 1 Focal length

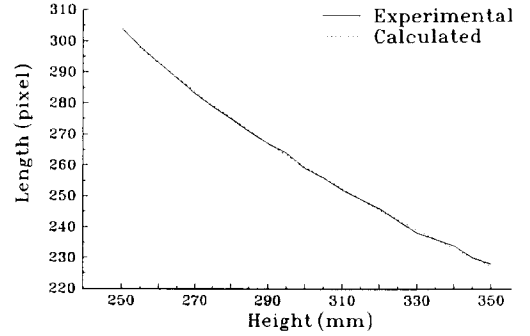
Range (mm)	D(pixel)	$z_c$ (mm)	Average error(pixel)
250 ≤ Height ≤ 350	1825.6	51.6	0.45
250 ≤ Height ≤ 600	1664.1	18.0	2.77



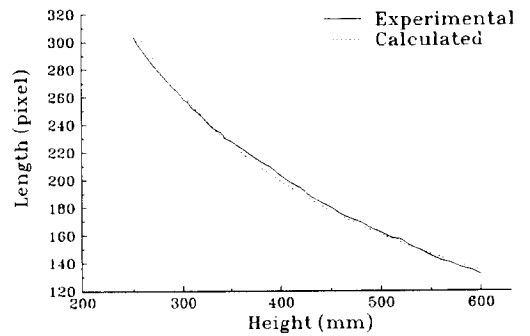
의 경우는 식 (4)에 의해 계산된 물체의 길이(점선)와 실험에 의해 측정된 길이(실선)가 거의 일치하여, 대부분의 측정범위에서 0.5 픽셀 이하였다. 오차가 1 픽셀 이상인 때는 250 mm의 높이에서 1.4 픽셀, 330 mm의 높이에서 1.1 픽셀인 두 경우뿐이었다. 그러나 카메라 높이를 250~600 mm로 넓게 변화시킨 경우에는 400 mm 부근에서 계산된 길이가 측정된 길이보다 최대 4.6픽셀까지 작았고, 250 mm 부근에서는 계산결과가 실험결과보다 무려 9.2 픽셀까지 크게 나타났다. 따라서 높이의 변화 폭이 작은 경우에 대해서는 이미지 상의 물체 길이로부터 높이오차를 계산하기 위해 원근투영법을 사용할 수 있음을 확인할 수 있다. 그러나 높이의 변화가 크게 되면 제한된 원근투영법에 복합렌즈 특성을 고려하는 보완대책이 필요하다.

4.2 POLPS에의 적용

Fig. 17과 Fig. 18은 각각 오프라인 프로그래밍 상에서 로봇의 위치와 자세가 보정되기 전과 보정된 후의 화면을 나타낸다. 두 그림은 육각기둥, 직육면체, 정육면체 형태의 물체로 구성된 작업환경에서 육각기둥의 단면을 기준도형으로 삼아 로봇의 자세를 보정하는 적용 예를 보여준다. 화면의 왼쪽



(a) Projection in narrow range



(b) Projection in wide range

Fig. 16 Result of perspective projection

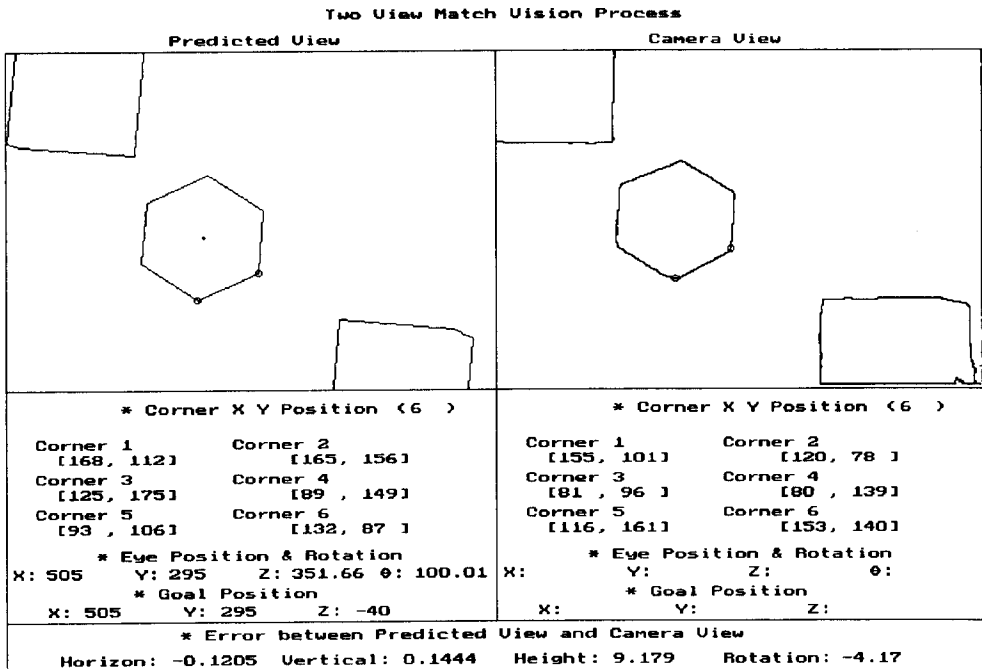


Fig. 17 Image comparison before calibration

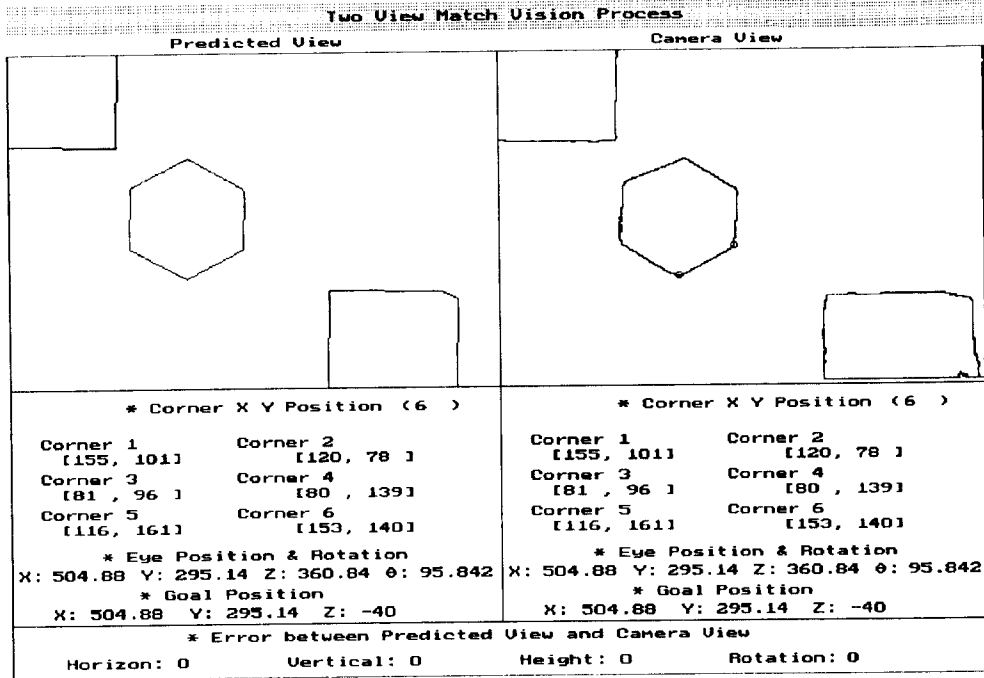


Fig. 18 Image comparison after calibration

상단에 예측이미지, 오른쪽 상단에 카메라이미지, 각 이미지 밑에 기준도형의 위치에 관련된 정보, 그리고 하단에는 두 이미지 사이의 오차가 제공된다. 각 이미지의 중앙에 육각각동이 위치하며, 그 단면은 반경이 1.5 cm인 원에 외접하는 정육각형이고 그 높이는 4 cm이다. 이미지의 왼쪽 상단에서 폭 4 cm, 길이 3 cm, 높이 2 cm인 직육면체의 일부를, 오른쪽 하단에서 각 변의 길이가 3 cm인 정육면체의 일부를 볼 수 있다. 이미지 아래에 육

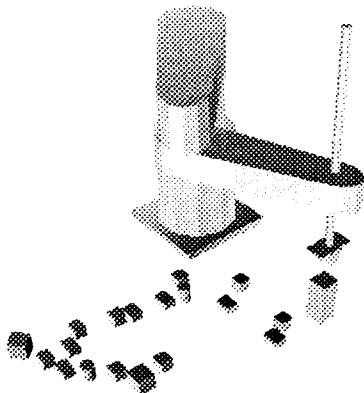


Fig. 19 Robot calibration on off-line programming

각각동의 단면에서 특징점으로 사용된 꼭지점의 좌표와 시각 위치와 관련된 자료가 제공되며, 그 밑에 본 논문에서 제안된 방법으로 계산된 병진오차, 높이오차, 회전오차 등이 제공된다.

보정 전의 단계에서 육각 단면의 이미지 비교에서 계산된 오차를 살펴보면, Fig. 17의 하단에서와 같이 도심의 위치오차는 화면에서 수평방향으로 -0.12 mm, 수직방향으로 0.14 mm, 그리고 높이방향으로 9.18 mm이며, 이 때 회전오차는 그 크기가 -4.17°이다. 보정 후에는 상기의 오차가 0으로 되면서 Fig. 18에서와 같이 두 이미지가 서로 일치하게 된다. Fig. 19는 POLPS 상에서 레적을 추종하는 작업에서 지정된 시간간격으로 위치를 보정하면서 로봇이 작동되는 화면을 나타낸다. 이와 같이 카메라에서 얻어진 시각정보를 이용하여, 오프라인 프로그래밍 상의 시뮬레이션 과정에서 실제 로봇에 대한 위치와 자세정보를 수시로 입력시키는 보정작업이 가능함을 보였다.

## 5. 결 론

본 논문에서는 오프라인 프로그래밍 상에서 계산

된 스카라 로봇의 위치와 자세를 보정하는 방법에 대해 다루었다. 시각정보의 처리에서 위치기반과 이미지기반 비주얼 서보잉의 장점을 이용함으로써 이미지의 비교가 신속하게 행해질 수 있는 기법을 마련하였다. 특히 단순 윤곽선도형의 특징점의 위치를 특징으로 사용함으로써 다양한 형태의 물체에 대해 제안된 방법이 쉽게 적용될 수 있도록 하였다. 원근투영법을 사용하여 카메라 높이와 이미지에 나타난 물체 길이 사이에 성립하는 간단한 관계식을 얻었다. 예측이미지와 카메라이미지 사이에 존재하는 오차의 계산에서 이 식을 적용함으로써 2차원 이미지의 비교로부터 3차원 오차를 산출하였으며, 동시에 보정의 실시간 적용가능성을 크게 높였다. 원근투영법의 타당성을 카메라를 이용한 실험으로 확인하여, 제안된 방법이 로봇의 보정에 유용하게 적용될 수 있음을 보였다. 특히 카메라 높이가 250~350 mm의 경우에는 여러가지 높이에서 실험한 결과 계산한 길이와 측정된 길이 사이의 차이가 1.0픽셀 미만으로 밝혀져 식의 실험성과 정확성이 확인되었다.

원근투영법을 이미지의 분석에 이용할 때 카메라 높이의 변화폭이 넓은 경우에는 정확성에 한계가 있다는 것이 발견되었다. 따라서 광범위한 카메라 높이에 대해 정확성을 유지하며 실시간으로 적용될 수 있는 이미지의 처리기법에 대한 연구<sup>(15)</sup>가 요망된다. 본 연구에서는 프로그래밍을 관장하는 컴퓨터와 스카라 로봇 사이에 인터페이스를 구축하는 과정<sup>(16)</sup>에서 로봇의 궤적추적을 지원하는 보정기법의 개발과 관련된 결과를 제시하였다. 향후 연구는 고속연산 신호처리칩(digital signal processor)을 이용하여 로봇의 실시간 보정을 실현하는데 초점을 맞추어 진행해 나갈 예정이다.

## 참고문헌

- (1) 범진환, 1996, "아아크 용접용 로봇의 오프라인 프로그래밍 응용을 위한 효과적인 캘리브레이션 방법 연구," 한국정밀학회지, 제13권, 제1호, pp 131~140.
- (2) Corke, P. I., 1993, "Visual Control of Robot Manipulators - A Review," *Visual Servoing* (Editor: Hashimoto, K.), World Scientific, pp. 1~31.
- (3) Westmore, D. B. and Wilson, W. J., 1991, "Direct Dynamic Control of a Robot Using an End-Point Mounted Camera and Kalman Filter Position Estimation," *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 2376~2384.
- (4) Hager, G. D., Chang, W. and Morse, A. S., 1994, "Robot Feedback Control Based on Stereo Vision: Towards Calibration Free Hand-Eye Coordination," *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 2850~2856.
- (5) Case, K. S., Jalkio, J. A. and Kim, R. C., 1987, "3-D Vision System Analysis and Design," *Three-Dimensional Machine Vision* (Editor: T. Kanade), KAP, pp. 63~95.
- (6) Yoshimi, B. H. and Allen, P. K., 1994, "Active, Uncalibrated Visual Servoing," *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 156~161.
- (7) 정창욱, 손 권, 1995, "오프라인 프로그래밍 시스템에서의 충돌회피 및 동적제어 시뮬레이션," 대한기계학회논문집, 제19권, 제10호, pp 2484~2494.
- (8) Donald, B. G., 1992, "Visual Tracking of Known Three-Dimensional Objects," *Int. J. of Computer Vision*, Vol. 3, pp. 243~270.
- (9) Gonzalez, R. C. and Woods, R. E., 1992, *Digital Image Processing*, Addison Wesley.
- (10) Craig, A. L., 1991, *Practical Image Processing in C*, John Wiley & Sons.
- (11) 신영수, 김현석, 1991, 3차원 그래픽: C언어로 배우는 이론과 알고리즘, 가남사.
- (12) Porter, K. and Floyd, M., 1990, *Stretching Turbo Pascal 5.5*, Brady.
- (13) Davies, E. R., 1990, *Machine Vision: Theory, Algorithm, Practicalities*, Academic Press.
- (14) Faires, J. D. and Burden, R. L., 1993, *Numerical Methods*, PWS Publishing Co.
- (15) Nelson, B. J. and Khosla, P. K., 1994, "The Resolvability Ellipsoid for Visual Servoing," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 156~161.
- (16) 이민철, 손 권, 이만형, 1995, "슬라이딩모드를 이용한 SCARA 로봇의 실시간 동적제어," 한국정밀공학회지, 제12권, 제13호, pp 113~121.