論文 90-27-11-5

영상특징을 이용한 로봇의 시각적 구동 방법

(Visual Servoing of an Eye-In-Hand Robot Based on Features)

張 源* 鄭 明 振* 卞 增 男*

(Won Jang, Myung Jin Chung, and Zeung Nam Bien)

要 約

본 논문에서는 시각정보에 의하여 로봇을 제어하기위해 영상으로부터 추출되는 feature를 이용하는 한 방법을 제안한다. 특별히 feature에 대한 수학적인 정의를 제안하였으며 로봇의 움직임과 feature vector의 미소한 변화 사이의 관계를 기술하였다. 이 과정에서 feature jacobian matrix와 그의 generalized inverse가 사용되었다. 로봇 자유도의 수보다 많은 feature를 사용하면 visual servoing의 성능을 향상시킬 수 있었다. 여러 예를 통하여, 본 논문에서 제안된 방법이 유효함을 보였다.

Abstract

This paper proposes a method of using image features in serving a robot manipulator. Specifically, the concept 'feature' is first mathematically defined and then the differential relationship between the robot motion and feature vector is derived in terms of Feature Jacobian Matrix and its generalized inverse. Also, by using more features than the number of DOFs of the robot, the visual servoing performance is shown to be improved. Via various examples, the method of feature-based servoing of a robot proposed in this paper is proved to be effective for conducting object-oriented robotic tasks.

I. 서 론

Flexible manufacturing의 구현을 위하여 그 주요구성요소인 로봇에게 주변환경에 유연하게 대응하는 능력을 부가하려는 연구가 여러 각도에서 시도되어 왔다. 시각기능을 통해 주위환경을 인지하여 그에 적당한 동작을 로봇 스스로가 결정하고 시행하도록 한다는 것은 비교적 일찍부터 많은 연구자들의 관심사였다. 그 결과로, '사전에 정밀하게 조정된 환경에서, calibration이 완료된 시각장치를 사용하여 원하는 물체의 3차원적 위치를 파악하고, base의 위치를 알고 있는 로봇에 명령을 내려서 원하는 작업을 수행한다'는 paradigm이 형성되기에 이르렀다. 그러나 '시각을 통한 물체의 3차원 위치 측정 algorithm'은 계산량의 과다, 그로 인한 through-put time의 연장, 상대적으로 큰 오차 등의 문제점을 갖고 있다. 또한 대부분 시각장치를 작업영역의 위에 장치하여 입면도를 촬영하는 형식을 취하므로, 로봇이 대상물을 취급하는 중에 시각장치의 시야를 가려버리는 상황이 발생한다.

*正會員, 韓國科學技術院 電氣및 電子工學科
(Dept. of Electrical Eng., KAIST)

接受日字 : 1990年 4月 13日

영장류가 물건을 보고 잡는데에 정밀한 위치 정보를 필요로 하지 않는 것으로 보이며, 특히 하등에 속하는 동물들이 먹이를 잘 잡는 것을 보면, '정밀한 위치계산'만이 시각을 이용한 제어의 유일한 방법은 아니라고 판단된다. 물론 생물체의 경우에는 기타 감각기들이 있으며, 출생 이후 계속 학습이 진행되어 왔다는 점을 무시할 수 없으나, 본 논문에서는 '시각을 통해 얻어지는 정보로부터 효율적으로 로봇의 운동을 결정'하는 문제인 visual servoing에 중점을 두고자 한다.

Visual servoing을 다룬 연구 발표가 상당수 있으나 그 중 대부분은 실질적으로는 '3차원 위치 측정'을 이용하고 그 기술형식을 달리한 것들이었다. 80년대에 들어와서 dynamic feature-based visual tracking system이 Sanderson 등에 의해 제안되었다.[2,3] 이 system은 로봇의 joint servo까지도 open-loop으로 변경한 다음 hand에 정착된 camera를 포함하는 하나의 closed-loop을 구성하고 적응제어방식을 채용하였다. 그러나 상용의 camera로는 joint servo가 요구하는 sampling time마다(보통 1msec이하) 영상을 촬영한다는 것이 불가능한 일이며, 계산량이 많이 요구되는 제어방식을 선택하였다는 것 등이 실제구현을 어렵게 하는 약점으로 지적된다. 대상물체의 정확한 CAD model이 주어진다는 가정하에 이 CAD model로 부터 얻을 수 있는 정보를 feature로 하여 vision-guided servoing system을 구현한 연구가 있었으나,[4,5] CAD model이 주어지지 않은 물체에 대해서는 융통성의 여지가 없으며, feature space와 camera space 사이의 Jacodian이 singular한 경우에 대한 대비책이 없다는 약점을 갖는다. CMAC을 이용한 학습제어가 채용되기도 하였으나,[6] 사전에 학습을 위한 반복시행이 많이 필요하다는 것이 단점으로 지적된다. 이 외에도 augmented image space[7]를 정의하고 이 space 에서 error를 evaluate하는 방식으로 visual servoing을 구현하는 등의 연구가 있었다.

본 논문에서 제안하는 visual servoing 방법은 촬영된 영상을 효율적으로 이용하여 eye-in-hand 로봇을 제어하는데 중점을 두고 있다. 영상과 관련된 'feature'에 대한 수학적인 정의를 제안하였으며, 제어방법의 설명과 함께 그 운용예를 보였다. 또한 로봇의 DOF보다 많은 수의 feature를 사용할 경우 visual servoing의 성능이 향상됨을 보였다.

Ⅱ. Feature의 정의와 이를 이용한 Robotic Task의 기술

Feature 또는 image feature라는 용어는 machine vision과 관련된 문헌에서는 흔히 찾아볼 수 있는 용어임에도 불구하고 워낙 함축적인 의미를 갖고 있는 까닭에 명백한 기술적인 정의를 내리지 못하고 있다. 보통은 영상의 어떤 특징을 일컫는데 사용되고 있으며, 물체인식과 같은 경우에는 binary image에 투영된 물체영상의 넓이, 무게중심의 위치와 같은 양들이 feature로서 이해되고 있다.

본 장에서는 feature를 수학적인 형태로 정의하고, feature라고 인식되고 있던 기존의 영상특징들중 상당부분을 이와 같은 형태로 표현하는 것이 가능함을 보인다. 또한 feature space에서 trajectory라는 형태로 robotic task를 기술하는 방법에 대해서 논한다.

1. Feature와 Feature Space

Camera를 통해 얻어지는 영상은(그림 1 참조) 입사광 휘도신호가 2차원적으로 배열된 것으로 볼 수 있으며,[2] (u, v)에서의 영상휘도를 $I(u, v)$, $(-U_0 \le u \le U_0, -V_0 \le v \le V_0)$라고 표시한다. $I(u, v)$의 형태는 촬영된 영상의 내용에 따라 결정된다.

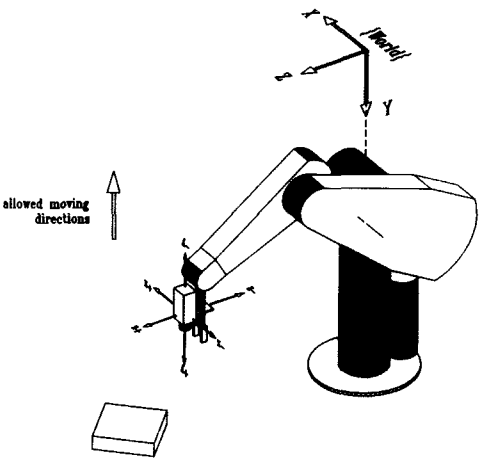


그림 1. Eye-In-Hand 로봇의 구성과 좌표계
Fig. 1. Configuration and coordinate frame for the eye-in-hand robot.

Feature f를 촬영된 영상으로부터 얻어지는 scalar 양으로서 다음과 같이 정의한다.

$$f = \int_{-V_0}^{V_0} \int_{-U_0}^{U_0} k(u, v, I(u, v))\, du\, dv \qquad (1)$$

여기서 kernel function $k(\cdot, \cdot, \cdot)$는 mapping으로서,

고려되는 영상특징의 성격에 따라 선형, 비선형이 좌우된다. 때로는 delta function을 포함하는 것도 가능하다. 제안된 정의식이 다양한 특징을 기술할 수 있음을 보이기 위해 자주 사용되는 몇몇 특징들[8,9]을 (1)식과 같은 형태로 기술하였다.

(i) 화면에 투영된 물체의 면적 :

$$f_{area}=\iint_{image} s(I(u,v)-I_{th})\,du\,dv \qquad (2)$$

여기서 $s(\cdot)$는 unit step function을 의미하며, I_{th}는 binarization시의 threshold level을 표시한다. 즉 $s(I(u,v)-I_{th})$가 'binarization'에 해당하는 transform을 행하는 것이다.

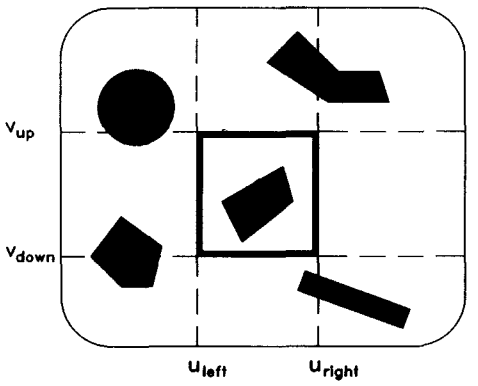


그림 2. 4개의 상수로 구성되는 윈도우
Fig. 2. A window defined by 4 constants.

여러 물체가 시야에 있거나 특정영역에 대해서만 관심이 있는 경우에는 그림 2에 보인 바와 같이 네 개의 상수 u_{left}, u_{right}, v_{down}, v_{up}로 window를 지정하여 사용할 수 있다. 지정된 window 내부에 있는 물체의 면적은 다음과 같이 기술할 수 있다.

$$f_{windowed\ area}=\iint_{image}\Big\{s(u-u_{left})-s(u-u_{right})\Big\}\cdot\Big\{s(v-v_{down})-s(v-v_{up})\Big\}\cdot s(I(u,v)-I_{th})\,du\,dv \qquad (3)$$

(ii) 투영된 물체영상에 대한 (p+q)차의 moment $f_{m,pq}$:

$$f_{m,pq}=\iint_{image} u^q v^p \cdot s(I(u,v)-I_{th})\,du\,dv \qquad (4)$$

투영된 물체영상에 대한 (p+q)차의 central moment $f_{\mu,pq}$:

$$f_{\mu,pq}=\iint_{image}(u-u_c)^p(v-v_c)^q\cdot s(I(u,v)-I_{th})\,du\,dv \qquad (5)$$

단, $u_c=\dfrac{f_{m,10}}{f_{m,00}}$, $v_c=\dfrac{f_{m,01}}{f_{m,00}}$

(iii) θ만큼 기울어진 직선에 대한 물체영상의 projection[8] :

$$P_\theta(\lambda)=\iint_{image} s(I(u,v)-I_{th})\cdot\delta(u\cos\theta+v\sin\theta-\lambda)\,du\,dv \qquad (6)$$

여기서 $\delta(\cdot)$는 Dirac delta function을 의미하며, λ는 직선 $u\cos\theta+v\sin\theta=\lambda$ 으로부터 원점까지의 최단거리를 의미한다. (그림 3 참조)

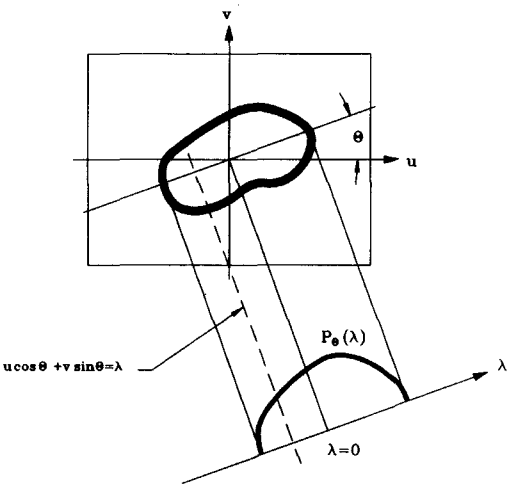


그림 3. θ만큼 기울어진 직선에 대한 영상의 프로젝션
Fig. 3. Projection of an image on the line with angle θ.

(iv) 영상(gray-level)의 Randon transform[9] :

$$f_{Randon}(\lambda,\theta)=g(\lambda,\theta)=\iint_{image} I(u,v)\,\delta(u\cos\theta+v\sin\theta-\lambda)\,du\,dv \qquad (7)$$

여기서 $g(\lambda,\theta)$는 ray-sum이라 불리우는 양으로서, θ만큼 기울고 원점으로부터 λ만큼 떨어진 직선을 따라 $I(u,v)$를 적산한 것에 해당한다.[9]

(v) Template $I_{temp}(u,v)$와 촬영된 영상 $I(u,v)$에 대해서 template matching을 하는 경우에, template가

(p, q)만큼 shift된 때의 mismatch 정도를 나타내는 mismatch energy $\sigma_\eta^2(p, q)$[9] :

$$f_{mismatch}(p, q)=\sigma_\eta^2(p, q)=\iint_{image}(I(u, v)-I_{temp}(u-p, v-q))^2\,du\,dv \tag{8}$$

Template와 영상사이의 cross-correlation :

$$f_{cross}(p, q)=\iint_{image}I(u, v)\,I_{temp}(u-p, v-q)\,du\,dv \tag{9}$$

(vi) 영상(gray-level)의 Fourier transformation :

$$R(w_1, w_2)+j\,I(w_1, w_2)=\int\int_{image}I(u, v)e^{-jw_1u}e^{-jw_2v}\,du\,dv \tag{10}$$

각 (w_1, w_2)쌍에 대해, 복소수인 $R(w_1, w_2)+jI(w_1, w_2)$이 결정된다.

생각할 수 있는 모든 영상특징을 (1)의 정의식으로 기술할 수 있는 것은 아니어서, chain code나, 여러 장의 영상으로부터 추출한 거리정보[10]를 기술하는 데에는 어려움을 겪는 것이 사실이지만, 예를 통해 알 수 있듯이 robot vision 분야에서 실질적으로 자주 사용되는 특징들의 상당부분에 대해서 기술이 가능하다.

촬영된 영상으로부터 여러 feature를 추출하여 사용하는 경우에 m-차원의 feature space F를 다음과 같이 정의한다.

$$F=\left\{f=(f_1, \cdots, f_m)^T\ f_i=\iint_{image}k_i(u, v, I(u, v))\,du\,dv \;\Big|\; i=1, \cdots, m\right\} \tag{11}$$

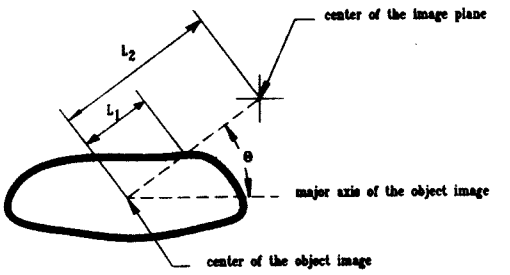
여기서 $(\cdot)^T$은 transpose를 의미한다. **F**의 element인 **f**에 대해서 norm을 다음과 같이 정의한다.

$$\|\mathbf{f}\|=\|(f_1, \cdots, f_m)^T\|=\left[\sum_{i=1}^{m}|f_i|^2\right]^{1/2} \tag{12}$$

Visual servoing을 위해서는 로봇의 DOF만큼의, 또는 그보다 많은 feature를 사용하는 것이 바람직스럽다.

2. Feature Space와 Robotic Task

Feature를 이용하여 servoing을 하는 경우에는, 로봇의 task를 feature의 time function $f(t), 0\le t\le T_{end}$로 기술하여 이용할 수 있다. 이 방법은 task의 내용적인 면을 묘사할 수 있어서 종전과 같이 via-point의 위치를 기술하는 방법보다 상황변화에 적응하기가 쉽다는 장점을 갖는다. 'feature의 time function'을 feature trajectory라 칭하며, m개의 feature를 사용하는 경우에 feature trajectory $\mathbf{f}(t)=(f_1(t), \cdots, f_m(t))^T,\ 0\le t\le T_{end}$는 feature space **F** 내의 time-assigned path에 해당한다.



* θ is assigned to the feature 1.
* The ratio of L_1 to L_2 is assigned to the feature 2.
* The area of the projected image of the object is assigned to the feature 3.

그림 4. 예제에서 선택된 feature들
Fig. 4. Features selected for the illustrative example.

예로서 eye-in-hend 로봇이 시야에 보이는 물체의 윤곽선을 따라 움직이도록 하는 'contouring task'[7]가 feature trajectory로 기술될 수 있음을 보인다. 이 task를 위해서 다음과 같이 정의되는 3개의 feature들을 정의하여 사용한다. 각 feature의 물리적 의미는 그림 4에 표현되어 있다.

$$f_1=\iint_{image}\mathrm{Atan2}(-v, -u)\,\delta(u-u_c, v-v_c)\,du\,dv \tag{13}$$

$$f_2=\iint_{image}k_{sub}(u, v)\,\delta(u-u_c, v-v_c)\,du\,dv \tag{14}$$

$$f_3=\iint_{image}s(I(u, v)-I_{th})\,du\,dv \tag{15}$$

여기서 $\mathrm{Atan2}(\cdot, \cdot)$는 4-quadrant arc tangent function[11]이고, $\delta(\cdot, \cdot)$는 two dimensional dirac delta function이다. 그외에 $u_c, v_c, k_{sub}(u, v)$의 의미는 다음과 같다.

$$u_c=\frac{\iint_{image}u\cdot s(I(\alpha\cdot\beta)-I_{th})\,d\alpha d\beta}{\iint_{image}s(I(\alpha, \beta)-I_{th})\,d\alpha\,d\beta}$$

$$v_c=\frac{\iint_{image}v\cdot s(I(\alpha, \beta)-I_{th})\,d\alpha\,d\beta}{\iint_{image}s(I(\alpha, \beta)-I_{th})\,d\alpha\,d\beta}$$

$$k_{sub}(u,v)=\frac{\iint_{image}(\frac{v}{u}\alpha-\beta)\,|s(\alpha-u)-s(\alpha)|\,d\alpha d\beta}{\iint_{image}\delta(\frac{v}{u}\alpha-\beta)s(I(\alpha,\beta)-I_{th})\,|s(\alpha-u)-s(\alpha)|\,d\alpha d\beta}$$

(when u=0)

$$=\frac{\iint_{image}\delta(\alpha)\,|s(\beta-v)-s(\beta)|\,d\alpha d\beta}{\iint_{image}\delta(\alpha)s(I(\alpha,\beta)-I_{th})\,|s(\beta-v)-s(\beta)|\,d\alpha\,d\beta}$$

(when u≠0)

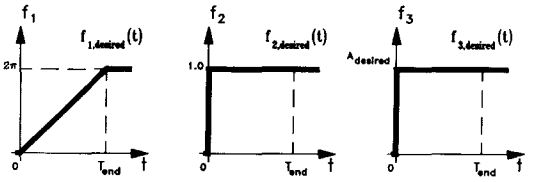
이상과 같이 정의된 feature들을 이용하여 'contouring task'를 다음과 같이 표현할 수 있다.

$$f_{1,desired}(t)=\frac{2\pi}{T_{end}}\cdot(s(t)-s(t-T_{end}))\cdot t\quad 0\le t\le T_{end} \tag{16}$$

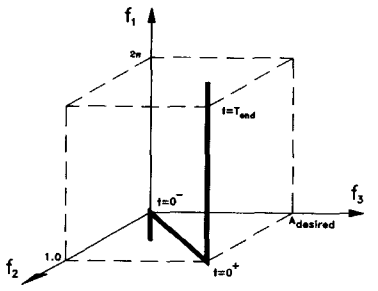
$$f_{2,desired}(t)=1.0\cdot s(t)\qquad 0\le t\le T_{end} \tag{17}$$

$$f_{3,desired}(t)=A_{desired}\cdot s(t)\qquad 0\le t\le T_{end} \tag{18}$$

이 feature trajectory를 feature space에서 표시된 것을 그림 5에 보였다. 이 방법의 장점으로는 어떤 형태의 물체가 시야에 들어오더라도 (16, 17, 18) 식의



(a)



(b)

그림 5. 'Contouring task'를 위한 feature 궤적
(a) 각 feature의 형태
(b) Feature 공간에서 본 feature 궤적

Fig. 5. Feature trajectory for the 'contouring task.'
(a) profile of each feature,
(b) feature trajectory in the feature space.

feature trajectory가 주어지면 보이는 물체의 윤곽선을 따라 'contouring task'가 수행된다는 점을 들 수 있다.

Ⅲ. Feature를 사용한 로봇의 운동제어

본 장에서는 앞 장에서 정의된 feature와 feature trajectory에 근거하여 로봇의 운동을 제어하는 문제를 다룬다. 우선 로봇의 미소한 움직임과 feature의 변화사이의 관계를 파악하고 이를 이용하여 로봇움직임을 제어하는 방법을 논한다.

일반적으로 로봇의 자유도는 n이고 그보다 많은 m개의 feature가 선택되었다고 가정한다. m 차원의 feature vector $(f_1,\cdots,f_m)^T$를 **f**로서 표기하고, 로봇 end-effector의 위치 $(x_1,x_2,\cdots,x_n)^T$를 **X**로서 표기한다. 로봇의 end-effector와 그에 정착된 camera 의 위치사이의 offset이 없는, gripper를 camera로 대체한 것과 같은 상황을 가정하였다.

영상에서 추출되는 feature에 근거하여 로봇의 움직임을 제어하는 과정은 다음과 같다.

[Step 1] 현재위치 $\mathbf{X}_i$에서 영상을 촬영하고 feature들을 추출한다. ($\mathbf{f}_{i,actual}$)

[Step 2] 원하는 값 $\mathbf{f}_{i,desired}$와 비교하여 오차 $\delta\mathbf{f}_i$가 있다면 ($\delta\mathbf{f}_i=\mathbf{f}_{i,desired}-\mathbf{f}_{i,actual}$), $\delta\mathbf{f}_i$를 없애는데 필요한 로봇의 움직임 $\delta\mathbf{X}_i$를 계산한다.

[Step 3] 로봇이 $\delta\mathbf{X}_i$만큼 움직이도록 한다.

[Step 4] Step1으로 돌아간다.(이상의 과정을 $\|\delta\mathbf{f}_i\|$가 소정의 기준치 이하가 될 때까지 반복한다. (i=1, 2, 3, ⋯))

본 장에서 중점을 두는 것은 $\delta\mathbf{X}$를 계산하는 방법이며 feature와 로봇 end-effector사이의 관계를 이용하고자 한다. (1)과 같이 정의되는 feature는 로봇의 end-effector에 정착된 camera를 통해 촬영된 영상으로부터 추출되는 것이므로 로봇 end-effector 가 움직임에 따라 촬영되는 영상이 달라지고 또한 추출되는 feature들의 값도 변화하게 된다. 이런 관점에서 'feature의 값 f는 로봇 end-effector의 위치 **X**의 영향을 받는다'고 말할 수 있으며 f를 **X**의 함수로 볼 수 있다.

$$f=G(x_1,\cdots,x_n)=G(\mathbf{X}) \tag{19}$$

m개의 feature를 사용하는 경우에는 m-dimensional feature vector $\mathbf{f}=(f_1,\cdots,f_m)^T$를 이용하여 다음과 같이 쓸 수 있다.

$$\mathbf{f}=\begin{bmatrix} f_1 \\ \cdot \\ \cdot \\ \cdot \\ f_m \end{bmatrix}=\begin{bmatrix} G_1(x_1,\cdots,x_n) \\ \cdot \\ \cdot \\ \cdot \\ G_m(x_1,\cdots,x_n) \end{bmatrix}=\mathbf{G}(\mathbf{X}) \qquad (20)$$

로봇 end-effector의 미소한 움직임을 $\delta\mathbf{X}=(\delta x_1,\cdots,\delta x_n)^T$이라하고, 그에 따르는 feature들의 변화를 $\delta\mathbf{f}=(\delta f_1,\cdots,\delta f_m)^T$라하면 식 (20)을 고려하여 다음과 같이 $\delta\mathbf{f}$와 $\delta\mathbf{X}$사이관계의 1차 근사식을 생각할 수 있다.

$$\delta\mathbf{f}\simeq\mathbf{J}_F\cdot\delta\mathbf{X} \qquad (21)$$

단,

$$\mathbf{J}_F=\begin{bmatrix} \frac{\partial G_1}{\partial x_1} & \cdots & \frac{\partial G_1}{\partial x_n} \\ & \cdot & \\ & \cdot & \\ & \cdot & \\ \frac{\partial G_m}{\partial x_1} & \cdots & \frac{\partial G_m}{\partial x_n} \end{bmatrix}$$

식 (21)의 m×n matrix는 로봇 end-effector의 움직임이 feature의 변화에 미치는 영향을 나타내는 것으로서 $\mathbf{J}_F$라 표시하고 feature jacobian matrix라는 이름으로 부르도록 한다.

로봇 움직임의 제어를 위해서는 $\delta\mathbf{f}(=\mathbf{f}_{desired}-\mathbf{f}_{actual})$가 주어진 상황에서 식 (21)을 풀어서 $\mathbf{f}_{actual}$을 $\mathbf{f}_{desired}$로 변화시킬 수 있는 로봇의 움직임 $\delta\mathbf{X}$를 구해야 한다. Feature의 갯수가 로봇의 DOF와 같다면,[2,3] $\mathbf{J}_F$의 inverse인 $\mathbf{J}_F^{-1}$을 써서 $\delta\mathbf{X}$를 $\mathbf{J}_F^{-1}\cdot\delta\mathbf{f}$로 계산할 수 있으나, 로봇의 DOF만큼의 feature 만을 사용해야 한다는 제약이 따른다. $\mathbf{J}_F$의 pseudo inverse를 사용하여 nonsquare인 $\mathbf{J}_F$에 대처한 연구가 있었으나[5] 항상 그 rank가 full일 것을 전제하고 있다. 실제상황에서는 계속 움직이면서 촬영한 영상으로부터 $\mathbf{J}_F$가 결정되므로 어떤 feature들이 작업과정 내내 $\mathbf{J}_F$의 full-rank를 유지해 줄 것인가를 보장하거나 예측하기가 쉬운 일이 아니다. 그러나 (1)과 같이 주어지는 수학적 정의식을 잘 이용하면 주어진 feature trajectory를 따라 움직이는 중에 $\mathbf{J}_F$가 singular하게 될 가능성의 여부정도는 판단할 수 있다.

우선 feature의 값이 실제로는 로봇 end-effector의 위치에 의해서도 영향을 받는다는 의미에서 $\mathbf{f}$의 i번째 feature f_i의 정의식을 다음과 같이 풀어쓸 수 있다.

$$f_i=f_i(x_1,\cdots,x_n)=\int_{-V_\theta(x_1,\cdots,x_n)}^{V_\theta(x_1,\cdots,x_n)}\int_{-U_\theta(x_1,\cdots,x_n)}^{U_\theta(x_1,\cdots,x_n)} k_i(u(x_1,\cdots,x_n),v(x_1,\cdots,x_n),I(u,z))\,du\,dv \qquad (22)$$

Kernel function의 형태에 따라서는 f_i가 $\mathbf{X}$의 간단한 함수형태가 되어 $\mathbf{J}_F$의 i-j항인 $\partial f_i/\partial x_j$가 쉽게 계산되기도 하지만, 일반적으로는 $\partial f_i/\partial x_j$는 다음과같은 형태를 갖는다.

$$\frac{\partial f_i}{\partial x_j}=\int_{-V_\theta(x_1,\cdots,x_n)}^{V_\theta(x_1,\cdots,x_n)}\Big(k_i(u(x_1,\cdots,x_n),V_\theta(x_1,\cdots,x_n),I(u,V_\theta))+k_i(u(x_1,\cdots,x_n),-V_\theta(x_1,\cdots,x_n),I(u,-V_\theta))\Big)du\cdot\frac{\partial V_\theta(x_1,\cdots,x_n)}{\partial x_j}$$
$$+\int_{-V_\theta(x_1,\cdots,x_n)}^{V_\theta(x_1,\cdots,x_n)}\Big(k_i(U_\theta(x_1,\cdots,x_n),v(x_1,\cdots,x_n),I(U_\theta,v))+k_i(-U_\theta(x_1,\cdots,x_n),v(x_1,\cdots,x_n),I(-U_\theta,v))\Big)dv\cdot\frac{\partial U_\theta(x_1,\cdots,x_n)}{\partial x_j}$$
$$+\int_{-V_\theta(x_1,\cdots,x_n)}^{V_\theta(x_1,\cdots,x_n)}\int_{-U_\theta(x_1,\cdots,x_n)}^{U_\theta(x_1,\cdots,x_n)}\frac{\partial}{\partial x_j}I(u(x_1,\cdots,x_n),v(x_1,\cdots,x_n),I(u,v))\,du\,dv \qquad (23)$$

$\mathbf{J}_F$의 singularity 가능성여부는 (23)과 같이 주어지는, kernel function의 편미분 column들간의 선형독립 여부를 시험해 보면 알 수 있다.[13] 이것은 수행하려는 task에 가장 알맞은 feature의 set을 결정하는데 있어서 하나의 기준이 될 수 있다.

실제 작업환경을 고려해보면 모든 조건을 만족 시키는 feature들이 충분하지 못한 경우가 있으므로 사용자의 feature 선택에 최대한으로 융통성을 허용하기 위해서는 $\mathbf{J}_F$가 singular하게 되더라도 적절한 대응움직임을 계산할 수 있어야 한다. 본 논문에서는 $\mathbf{J}_F$의 generalized inverse[12]인 $\mathbf{J}_F^+$를 사용하여 필요한 움직임 $\delta\mathbf{X}$를 다음과 같이 계산하였다.

$$^B\delta\mathbf{X}_0=\mathbf{J}_F^+\cdot\delta\mathbf{f} \qquad (24)$$

$\mathbf{J}_F$가 full-rank이면 $\mathbf{J}_F$는 $(\mathbf{J}_F^T\mathbf{J}_F)^{-1}\mathbf{J}_F^T$가 된다. $\mathbf{J}_F$의 rank가 full이 아닌 경우에는 generalized inverse[12] $\mathbf{J}_F^+$를 사용하게 되는데 이렇게 얻어진 $\delta\mathbf{X}_0$는 다음과 같은 의미에서 식 (21)의 best approximate solution에 해당한다.[12]

1) 모든 $\delta\mathbf{X}\in\mathbf{R}^n$에 대해서 다음이 성립한다

$$(\mathbf{J}_F\cdot\delta\mathbf{X}-\delta\mathbf{f})^T\cdot(\mathbf{J}_F\cdot\delta\mathbf{X}-\delta\mathbf{f})\geq(\mathbf{J}_F\cdot\delta\mathbf{X}_0)^T\cdot(\delta\mathbf{X}_0-\delta\mathbf{f})$$

2)

$$(\mathbf{J}_F\cdot\delta\mathbf{X}-\delta\mathbf{f})^T\cdot(\mathbf{J}_F\cdot\delta\mathbf{X}-\delta\mathbf{f})=(\mathbf{J}_F\cdot\delta\mathbf{X}_0-\delta\mathbf{f})^T\cdot(\mathbf{J}_F\cdot\delta\mathbf{X}_0-\delta\mathbf{f}),$$

인 $\delta\mathbf{X}$에 대해서는 $\delta\mathbf{X}\neq\delta\mathbf{X}_0$라면 다음이 성립한다.

$$\delta\mathbf{X}^T\cdot\delta\mathbf{X}>\delta\mathbf{X}_0^T\cdot\delta\mathbf{X}_0$$

다음에 $\mathbf{J}_F$의 실제적인 몇가지 예를 보인다.

[예 1]

우선 그림 6에 보인 간단한 경우에 대하여 논한다. Tesk는 투영된 물체의 면적이 원하는 값이 되도록 y축을 따라 로봇의 end-effector의 위치를 조정하는 것이다. 이는 로봇의 DOF가 1인 경우에 해당하며,

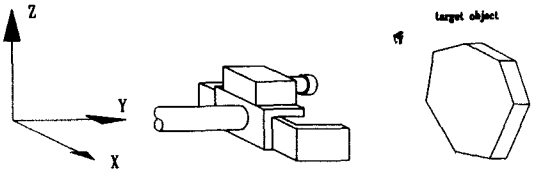


그림 6. Feature를 이용한 eye-in-band 로봇의 제어
Fig. 6. Control of on eye-in-hand robot using a feature/features.

feature인 gray-level image에서의 물체의 면적을 다음과 같이 기술할 수 있다.

$$f_{area,gray}=\iint_{image} I(u,v)\,du\,dv \tag{25}$$

$y_1=y_{cam}$으로서 로봇 end-effector에 장착된 camera의 위치를 표시하고 y_{obj}로는 목표물체의 위치를 표시한다. 이 예와 같은 경우에는 로봇 end-effector의 위치와 feature와의 관계를 기하학적인 관찰을 통하여 다음과 같이 직접 구할 수 있다.

$$f_{area}=G(y_{cam})=\frac{Af^2}{K^2}\,\frac{1}{(y_{obj}-y_{cam}+f)^2} \tag{26}$$

여기서 f는 camera의 focal length, K는 camera의 종류에 따라 결정되는 scale factor, A는 대상물체의 실제넓이이다. 이때 δf_{area}와 δy_{cam} 사이의 관계는 다음과 같이 계산된다.

$$\delta f_{cam}=\mathbf{J}_F\cdot\delta y_{cam}=\frac{\partial G}{\partial y_{cam}}\,\delta y_{cam} \tag{27}$$

단,

$$\frac{\partial G}{\partial y_{cam}}=\frac{2Af^2}{K^2}\,\frac{1}{(y_{obj}-y_{cam}+f)^3}=\frac{2K}{Af}(f_{area})^{3/2} \tag{28}$$

이 예에서는 $\mathbf{J}_F$의 dimension은 1×1에 해당하며, $\mathbf{J}_F^+$는 다음과 같이 계산된다.

$$\mathbf{J}_F^+=\left[\frac{\partial G}{\partial y_{cam}}\right]^{-1}=\frac{Af}{2K}(f_{area})^{-3/2} \tag{29}$$

[예 2]

이번에는 좀 더 복잡한 형태로 정의되는 feature들을 사용한다. 로봇은 x축, y축, z축 방향으로 3DOF를 갖는다고 가정한다. 사용되는 feature들의 정의는 다음과 같다.

$$f_1=\iint_{image} u\cdot\delta(u-u_c)\,du\,dv \tag{30}$$

$$f_2=\iint_{image} v\cdot\delta(v-v_c)\,du\,dv \tag{31}$$

$$f_3=\iint_{image} I(u,v)\,du\,dv \tag{32}$$

여기서 u_c, v_c의 의미는 다음과 같다.

$$u_c=\frac{\iint_{image}\alpha\cdot s(I(\alpha,\beta)-I_{th})\,d\alpha\,d\beta}{\iint_{image} s(I(\alpha,\beta)-I_{th})\,d\alpha\,d\beta}$$

$$v_c=\frac{\iint_{image}\beta\cdot s(I(\alpha,\beta)-I_{th})\,d\alpha\,d\beta}{\iint_{image} s(I(\alpha,\beta)-I_{th})\,d\alpha\,d\beta}$$

$(x_1, x_2, x_3)=(x_{cam}, y_{cam}, z_{cam})$으로 로봇 end-effector에 정착된 camera의 위치를 표시하고, $(x_{obj}, y_{obj}, z_{obj})$로서 대상물체의 위치를 표시한다. 정의된 feature들의 값과 camera의 위치사이의 관계를 기하학적인 관찰을 통해 다음과 같이 얻을 수 있다.

$$f_1=G_1(x_{cam}, y_{cam}, z_{cam})=\frac{f}{K}\,\frac{x_{obj}-x_{cam}}{y_{obj}-y_{cam}+f} \tag{33}$$

$$f_2=G_2(x_{cam}, y_{cam}, z_{cam})=\frac{f}{K}\,\frac{z_{obj}-z_{cam}}{y_{obj}-y_{cam}+f} \tag{34}$$

$$f_3=G_3(x_{cam}, y_{cam}, z_{cam})=\frac{Af^2}{K^2}\cdot\frac{1}{(y_{obj}-y_{cam}+f)^2} \tag{35}$$

여기서 f는 camera의 focal length, K는 camera의 종류에 따라 결정되는 scale factor, A는 대상물체의 실제면적이다.

$\delta\mathbf{f}$와 $\delta\mathbf{X}_{cam}$ 사이의 관계는 3×3의 Feature jacobian $\mathbf{J}_F$를 사용하여 다음과 같이 기술할 수 있다.

$$\delta\mathbf{f}=\begin{bmatrix}\delta f_1\\ \delta f_2\\ \delta f_3\end{bmatrix}=\mathbf{J}_F\cdot\delta\mathbf{X}_{cam}=\begin{bmatrix}\frac{\partial G_1}{\partial x_{cam}} & \frac{\partial G_1}{\partial y_{cam}} & \frac{\partial G_1}{\partial z_{cam}}\\ \frac{\partial G_2}{\partial x_{cam}} & \frac{\partial G_2}{\partial y_{cam}} & \frac{\partial G_2}{\partial z_{cam}}\\ \frac{\partial G_3}{\partial x_{cam}} & \frac{\partial G_3}{\partial y_{cam}} & \frac{\partial G_3}{\partial z_{cam}}\end{bmatrix}\cdot\begin{bmatrix}\delta x_{cam}\\ \delta y_{cam}\\ \delta z_{cam}\end{bmatrix} \tag{36}$$

Feature jacodian matrix의 각 element들은 로봇의 위치에 의한 영향을 받는 것이 보통이며 본 예의 경우에는 그 관계식을 식(33, 34, 35)으로부터 다음과 같이 유도할 수 있다.

$$\frac{\partial G_1}{\partial z_{cam}}=\frac{\partial G_2}{\partial x_{cam}}=\frac{\partial G_3}{\partial x_{cam}}=\frac{\partial G_3}{\partial z_{cam}}=0 \tag{37}$$

$$\frac{\partial G_1}{\partial x_{cam}} = \frac{f}{K(y_{obj}-y_{cam}+f)} = -\frac{1}{A^{1/2}}\cdot f_3^{1/2} \quad (38)$$

$$\frac{\partial G_1}{\partial y_{cam}} = \frac{f(x_{obj}-x_{cam})}{K(y_{obj}-y_{cam}+f)^2} = \frac{1}{A^{1/2}f}\cdot f_1\cdot f_3^{1/2} \quad (39)$$

$$\frac{\partial G_2}{\partial y_{cam}} = \frac{f(z_{obj}-z_{cam})}{K(y_{obj}-y_{cam}+f^2)} = \frac{K}{A^{1/2}}\cdot f_2\cdot f_3^{1/2} \quad (40)$$

$$\frac{\partial G_3}{\partial z_{cam}} = -\frac{f}{(K(y_{obj}-y_{cam}+f)} = -\frac{1}{A^{1/2}}\cdot f_3^{1/2} \quad (41)$$

$$\frac{\partial G_3}{\partial y_{cam}} = \frac{2Af^2}{K_2}\cdot\frac{1}{(y_{obj}-y_{cam}+f)^3} = \frac{2K}{Af}\cdot f_3^{3/2} \quad (42)$$

이상의 관계식을 이용하면, 매번 촬영되는 영상으로부터 추출된 feature값을 이용하여 $\mathbf{J}_F$를 on-line으로 계산할 수 있다. (단, f, K및 A는 사전에 주어지며 task 수행중에는 변화하지 않는다고 가정한다.)

$$\mathbf{J}_F \begin{bmatrix} -\frac{1}{A^{1/2}}f_3^{1/2} & \frac{K}{A^{1/2}f}f_1 f_3^{1/2} & 0 \\ 0 & \frac{K}{A^{1/2}f}f_2 f_3^{1/2} & -\frac{1}{A^{1/2}}f_3^{1/2} \\ 0 & 0 & \frac{2K}{Af}f_3^{1/2} \end{bmatrix}$$

$\mathbf{J}_F^+$ 는 다음과 같이 계산된다.

$$\mathbf{J}_F^+ = \begin{bmatrix} -A^{1/2}\frac{f_2}{f_3^{1/2}} & A^{1/2}\frac{f_1}{f_2 f_3^{1/2}} & \frac{Af}{2K}\frac{f_1}{f_2 f_3^{3/2}} \\ 0 & \frac{A^{1/2}f}{K}\frac{1}{f_2 f_3^{1/2}} & \frac{Af^2}{2K^2}\frac{1}{f_2 f_3^{3/2}} \\ 0 & 0 & \frac{Af}{2K}\frac{1}{f_3^{1/2}} \end{bmatrix} \quad (44)$$

Ⅳ. 응용 예

본 장에서는 feature와 feature trajectory를 사용하여 로봇 end-effector의 위치를 제어하는 예를 보인다.

[예 1]

이 예에서는 eye-in-hand 로봇에 있어서 feature error를 이용하여 point-to-point(PTP)제어를 행하는 것을 보인다. 로봇은 그림 6의 x축, y축 방향의 2 DOF를 가지며 대상물체는 길이가 ℓ_0 mm인 막대로서 camera의 영상평면과는 θ_0도 만큼 기울어져 있다고 가정한다. 앞의 예에서와 같이 로봇 end-effector의 위치와 장착된 camera의 위치가 일치한다고 가정하고, camera의 고개숙임이나 회전 등이 없어서 막대의 영상이 화면의 수직 중앙에 수평 선분으로 나타난다고 가정하였다. 또한 조명조건이 좋아서 배경과 대상물체가 쉽게 구분되는 상황을 가정하였다. 수행하고자 하는 과제는 주어진 위치에서 로봇이 출발하여 feature들이 원하는 값을 갖는 목표위치에 도달하도록 하는 것이다. Feature f_1으로는 막대의 투영된 길이를, f_2로는 막대영상의 무게 중심의 u좌표를 선택하였다. 이상의 feature들을 다음과 같이 기술

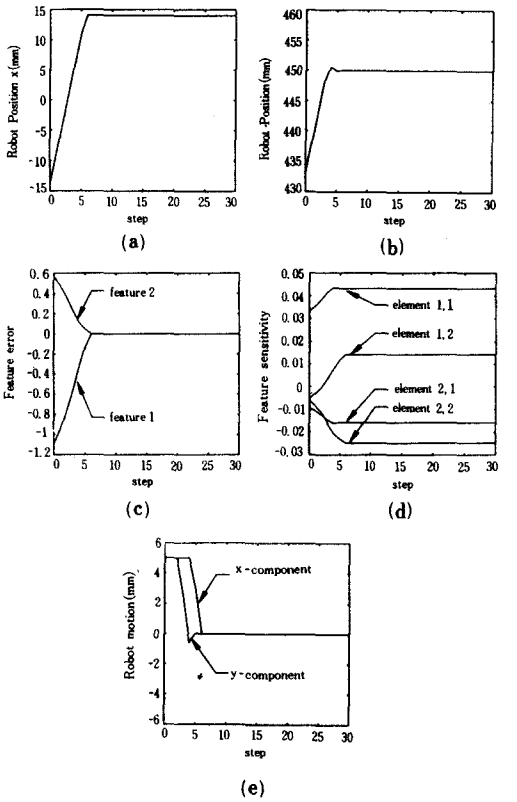


그림 7. 예제 1의 결과
(a) 로봇의 위치(X좌표)
(b) 로봇의 위치(Y좌표)
(c) Feature error
(d) Feature의 민감도
(e) 로봇의 운동

Fig. 7. Results of example 1.
(a) robot position, (X-coordinate)
(b) robot position, (Y-coordinate)
(c) feature error,
(d) feature sensitivity,
(e) robot motion.

할 수 있다.

$$f_1 = \iint_{Image} u \cdot \delta(u - u_c, v)\, du\, dv \qquad (45)$$

$$f_2 = \iint_{Image} \delta(u) \cdot s(I(u, v) - I_{th})\, du\, dv \qquad (46)$$

단,

$$u_c = \frac{\iint_{Image} \alpha \cdot \delta(\beta) \cdot s(I(\alpha \cdot \beta) - I_{th})\, d\alpha\, d\beta}{\iint_{Image} \delta(\beta) \cdot s(I(\alpha \cdot \beta) - I_{th})\, d\alpha\, d\beta}$$

출발점과 목표 feature값을 변화시키면서 다양한 simulation을 행하였으며, 그림 7은 그중 전형적인 한 결과를 보인다. 출발점은 Cartesian좌표계에서 (−14, 433)에 해당하고, 목표점은 $(f_1, f_2) = (-0.6892, 1.1388)$에 해당하는 곳이다. 그림 7의 d를 보면 수행도중에도 $\mathbf{J}_F$ element들의 크기 및 상대적인 비율이 계속 변화되고 있음을 알 수 있다. Gain을 계속 update하고 **f**와 $\mathbf{X}_{cam}$ 사이의 계속 변화되는 결합성도 반영할 수 있다는 점이 'on-line으로 $\mathbf{J}_F$를 개선시켜서 사용하는' 본 방법의 특징이라 할 수 있다.

[예 2]

이 예에서는 로봇의 DOF보다 많은 수의 feature (redundant feature set이라 칭함)를 사용함으로서 얻어지는 효과를 보인다. 대상으로서는 그림 8과 같은 시스템을 가정한다. 수행하려는 task는 eye-in-hand 로봇이 점선으로 표시된 desired trajectory 를 추적(follow)하려는 것이다. 역시 로봇은 x축, y축방향의 2 DOF를 갖는다고 가정하였다. Feature 로서는 각 guide object의 무게중심의 u좌표를 선택하였다. 그 정의식에 대해서는 식 (46)을 참조할 수 있다. Guide object 하나로부터 하나씩의 feature가 얻어지며, 로봇의 DOF만큼의 수효로부터 시작하여 feature

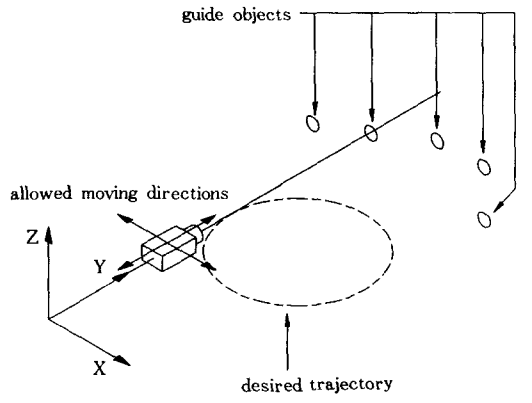


그림 8. Redundant Feature set을 이용한 시각서난 (예제 2 관련)

Fig. 8. Visual servoing via redundant feature set. (example 2)

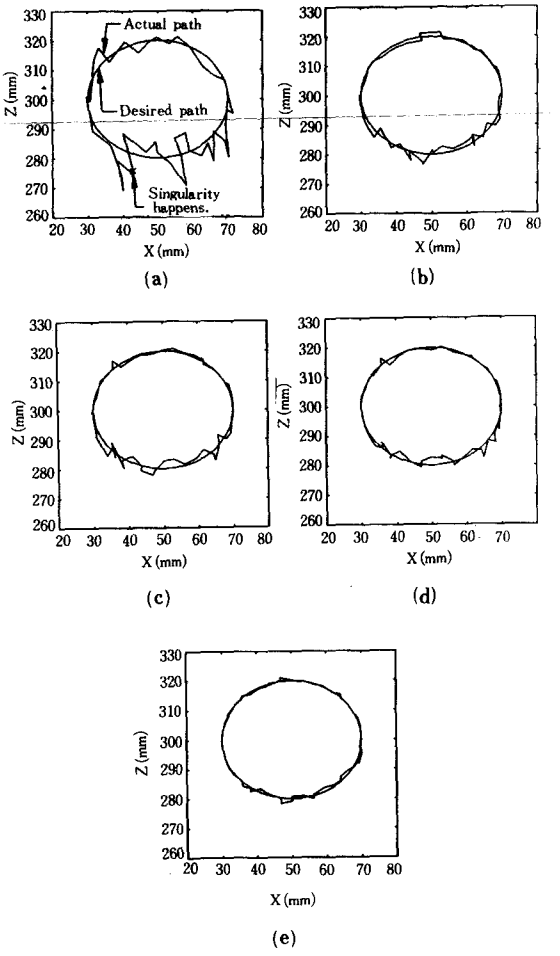


그림 9. Redundarct feature set을 통한 시각서보 성능의 개선효과
(a) (feature의 수) = (로봇 자유도의 수)
(b) (feature의 수) = (로봇 자유도의 수) + 1
(c) (feature의 수) = (로봇 자유도의 수) + 2
(d) (feature의 수) = (로봇 자유도의 수) + 3
(e) (feature의 수) = (로봇 자유도의 수) + 4

Fig. 9. Performance of visual servoing improved by redundant features.
(a) (# of features) = (# of DOF)
(b) (# of features) = (# of DOF) + 1
(c) (# of features) = (# of DOF) + 2
(d) (# of features) = (# of DOF) + 3
(e) (# of features) = (# of DOF) + 4

의 수를 늘여가면서 task를 수행시켜 보았다. Simulation 결과중 해상도가 256×256인 상황에 대한 결과를 그림 9에 보였다. 그림 9의 a중 'x'로 표시된 곳에서 J_F가 singular하게 되었으나 generalized inverse를 사용한 효과로 task를 끝까지 수행할 수 있었다.

V. 결 론

본 논문에서는 feature에 대한 수학적인 정의를 제안하고 feature를 이용한 visual servoing의 구현방법을 제안하였다. 제안된 방법은 feature jacobian matrix와 그 generalized inverse를 사용하며, 사전반복학습이 필요하지 않고 요구되는 계산량이 적다는 장점을 갖는다. Feature space에서 feature trajectory로 로봇의 DOF보다 많은 수의 feature를 사용하면 visual servoing의 성능을 더욱 개선할 수 있음을 보였다.

參 考 文 獻

[1] Loughlin, Clive and Ed Hudson, "Eye in Hand Robot Vision," in *Proceedings of The 2nd International Conference on Robot Vision and Sensory Controls*, pp. 263-270, IFS Ltd., Nov. 1982

[2] Sanderson, Arthur C. and Lee E. Weiss, "Adaptive Visual Servo Control of Robots," in *Robot Vision*, ed. Alan Pugh, pp. 107-116, IFS Ltd., 1983.

[3] Weiss, Lee E., Arthur C. Sanderson, and Charles P. Neuman, "Dynamic Sensor-Based Control of Robots with Visual Feedback," *IEEE Journal of Robotics and Automation*, vol. 3, no. 5, pp. 404-417, IEEE, Oct. 1987.

[4] Feddema, John T., C. G. G. Lee, and O. R. Mitchell, "Automatic Selection of Image Features for Visual Servoing of a Robot Manipulator," *in Proceedings 1989 IEEE International Conference on Robotics and Automation*, pp. 832-837, IEEE, May, 1989.

[5] Feddema, John T. and Owen R. Mitchell, "Vision-Guided Servoing with Feature-Based Trajectory Generation," *IEEE Transactions on Robitcs and Automation*, vol. 5, no. 5, pp. 691-700, IEEE, Oct. 1989.

[6] Miller, W. Thomas III, "Sensor-Based Control of Robotic Manipulators Using a General Learning Algorith," *IEEE Journal of Robotics and Automation*, vol. 3, no. 2, pp. 157-165, Apr. 1987.

[7] Jang, W., K.J. Kim, M.J. Chung, Z. Bien, "Concepts of Augmented Image Space and Transformed Feature Space for Efficient Visual Servoing of an Eye-In-Hand robot," accepted for publication in ROBOTICA

[8] Rosenfeld, Azriel, Avinash C. Kak, *Digital Picture Processing*, Academic Press, 1982.

[9] Jain, Anil K., *Fundamentals of Digital Image Processing*, Prentice-Hall, 1989.

[10] Jang, W., K.J. Kim, M.J. Chung, Z. Bien, "3-D Position Estimation for Eye-in-Hand Robot Vision," in *Proceedings of the 1988 Korean Automatic Control Conference*, (Seoul, Korea), Oct. 1988.

[11] Craig, John J., *Introduction to Robotics*, Addison-Wesley, 1986.

[12] Graybill, Franklin, A., *Introduction to Matrices with Applications in Statistics*, Wadsworth Publishing Company, 1969.

[13] Chen, Chi-Tsong, *Linear System Theory and Design*, CBS College Publishing, 1984.

著 者 紹 介

張 源 (正會員) 第26卷 第8號 參照
현재 한국과학기술원 전기및 전자공학과 박사과정

●

鄭 明 振 (正會員) 第26卷 第8號 參照
현재 한국과학기술원 전기및 전자공학과 교수

卞 增 男 (正會員) 第27卷 第1號 參照
현재 한국과학기술원 전기및 전자공학과 교수