

칼만필터를 이용한 이동 목표물의 실시간 시각추적의 구현 (The Implementation of the Realtime Visual Tracking of Moving Target by using Kalman Filter)

임 양 남, 방 두 열(전북대 대학원), 이 성 철(전북대 기계공학과)
Yang-Nam LIM, Du-Yol PANG(Chonbuk Univ.), Seong-Cheol LEE(Chonbuk Univ.)

ABSTRACT

In this paper, we proposed realtime visual tracking system of moving object for 2D target using extended Kalman Filter Algorithm. A targeting marker are recognized in each image frame and positions of target object in each frame from a CCD camera while the targeting marker is attached to the tip of the SCARA robot hand. After the detection of a target coming into any position of the field-of-view, the target is tracked and always made to be located at the center of target window. Then, we can track the moving object which moved in inter-frames. The experimental results show the effectiveness of the Kalman filter algorithm for realtime tracking and estimated state value of filter, predicting the position of moving object to minimize an image processing area, and by reducing the effect by quantization noise of image.

Keywords : Kalman filter, Real time visual tracking, Moving target, Image processing

1. 서 론

이동물체를 추적하는 연구에 화상정보를 많이 사용하고 있다. 화상정보를 이용하여 주변환경을 인식할 경우, 가장 문제가 되는 것은 정보의 계산량에 따른 실시간 처리의 어려움이다. 화상정보를 이용한 추적기법이 전부터 연구되어 왔으며, 화상정보를 이용한 위치예측과 이동물체의 추적에 계산량 증가로 인하여 실시간 추적의 어려움과 함께 화상대 이타의 양자화 및 노이즈 영향으로 계속 정도가 저하하게 된다. 그러므로 이동물체의 식별과 위치추적에 계산량이 많으므로 보다 간단하고 신속히 처리하는 알고리즘이 필요하다.

시각추적은 위치를 근거로 한 방법(position-based)과 화상을 근거로 한 방법(image-based)으로 구분되며, 추적 알고리즘으로 Wang⁽¹⁾ 등은 임의로 움직이는 물체의 상대위치와 자세를 추정하고 로봇을 제어하기 위하여 칼만필터를 이용하였고, Allen⁽²⁾은 optical flow 알고리즘을 이용하여 물체의 3차원 위치 정보를 추정하였다. 또, Papanikolopoulos⁽³⁾는 PID, LQG, Pole placement 등 다양한 제어법칙을 이용하여 시각 추적 시스템을 구현하기도 하였다. 이동물체 추적 연구에서 칼만필터는 순환적 구조를 가지고 있으므로 알고리즘이 단순하고 수렴속도가 빨라서 실시간 처리가 용이하며, 주파수영역이 아닌 시간영역에서 직접적으로 데이터 처리가 가능하므로 널리 사용하고 있다.

화상 입력보드에서 이미지 처리는 한개의 필드 화상을 2차화하고 중심산출 처리에 출력지연과 함께 양자화의 분해능 저하가 있다. 특히, 이전화면을 처리할 경우에는 이미지 보드로부터 다음 화상을 취하여 컴퓨터로 데이터의 전송 종료까지는 컴퓨터 CPU가 무부하 상태에 있는 시간이 길어짐으로 이의 활용이 요구된다.

따라서, 본 논문에서는 이미지 습득과 화상정보 전송 시작까지에 무부하 상태에 있는 CPU를 활용하여 알고리즘 연산을 수행하는 병렬처리 방법을 도입함으로 영상처리 시간을 최소화 시켜 퍼스널 컴퓨터의 이용과 칼만필터 알고리즘을 사용한 이동물체의 실시간 추적의 가능성을 고찰하였다.

2. 칼만필터 알고리즘

칼만필터는 순환적필터 방식으로 알고리즘이 단순하고 수렴성이 좋으며, 이전의 스펙트럼 분석 방식에서 벗어나 시간영역에서 직접적인 처리가 가능하므로 디지털 컴퓨터에서 사용하기에 적합한 방법이다. 그래서 현대제어, 신호처리, 우주항공분야 등에서 널리 적용되는 알고리즘으로 백색 가우스 잡음을 갖는 시스템에 있어서 최적의 필터이며, 현재와 과거 측정치, $[Y(1), \dots, Y(k)]$ 로 부터 정확한 시스템 상태변수 $X(k)$ 의 추정치를 순환적으로 얻는 알고리즘이다.

일반적으로 표적의 이동은 직각좌표계를 적용할 때 각 좌표축 방향에 대하여 다음과 같은 선형 운동방정식으로 모델링된다.

$$X(k) = AX(k-1) + w(k-1) \quad (1)$$

$$Y(k) = CX(k) + v(k) \quad (2)$$

여기서, X 는 운동모델의 상태변수, A 는 모델의 천이행렬(State Transition Matrix), Y 는 출력변수, 그리고 w, v 는 평균이 0이고 각각 공분산 Q, R 로 표시되는 독립적인 백색잡음 벡터들이다.

칼만필터 방정식의 결과식은 다음과 같이 표현된다.

Predictor equation

$$\hat{x}(k+1/k) = A\hat{x}(k/k-1) + K(k)[y(k) - C\hat{x}(k/k-1)] \quad (3)$$

Predictor(Kalman) Gain

$$K(k) = AP(k/k-1)C^T [CP(k/k-1)C^T + R(k)]^{-1} \quad (4)$$

Prediction mean square error

$$P(k+1/k) = [A - K(k)C]P(k/k-1)A^T + Q(k) \quad (5)$$

$P(k)$ 는 예측오차의 공분산을 표시하는 행렬로 최상의 추정치 $P(k)$ 를 최소화시키는 예측기이다. 위의 칼만필터 결과식에는 아래와 같은 가정들이 요구된다.

$$\begin{aligned} E[w(k)] &= 0, & E[v(k)] &= 0 \\ E[w(k)w(j)^T] &= Q(k)\delta(k,j) \\ E[v(k)v(j)^T] &= R(k)\delta(k,j) \end{aligned} \quad (6)$$

여기서, $E[\cdot]$ 은 평균값, $\delta(k, j)$ 는 Kronecker Delta함수이다.

본 논문에서 적용한 시스템 모델에 대한 칼만필터의 구조를 Fig.1에 나타내고 있으며, Fig.2는 칼만필터 알고리즘의 연산 Timing 선도를 보이고 있다.

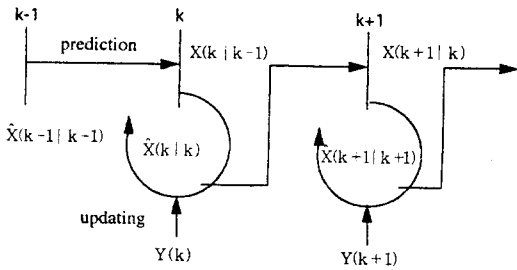


Fig.2 Timing diagram of Kalman Filter

3. 이미지 프로세싱

외부에 있는 임의의 한 점을 카메라로 촬영하였을 때, 이 한 점이 카메라의 영상면 어느 부분에 투시되는가를 알기 위해서는 외부의 점과 카메라 사이의 투시 관계를 알아야 한다. 즉, 카메라의 이미지로 부터 3차원 공간상의 물체의 위치를 알기 위하여 좌표 calibration이 필요하다. 이 보정은 3차원 공간상의 임의의 좌표나 시각센서에서 형성되는 2차원 영상좌표 사이의

대응관계를 구하는 것이다. 이미지 좌표와 실제 3차원 좌표사이의 관계식은 다음과 같이 구할 수 있다.

카메라의 초점거리를 f , 물체의 상하방향을 y 좌표, 좌우방향을 x 좌표라 할 때, 카메라에서 물체까지의 거리 z 는 이미 알고 있으므로 삼각형의 닮음과 화상상의 물체중심 위치 좌표 x_i, y_i 를 이용하여 다음 식으로 구할 수 있다.

$$x = \frac{x_i}{f} z, \quad y = \frac{y_i}{f} z \quad (7)$$

또한, 이동물체를 추적하기 위해서 본 논문에서는 추적 윈도우를 사용하였다. 목표물 추적은 임의의 시간 $t(k)$ 까지의 측정데이터가 주어졌을 때 시간 $t(k)$ 에서의 목표물의 상태 즉, 위치, 속도, 가속도 등을 계산하여 $t(k+1)$ 시점에서의 목표물의 상태를 예측하고, 예측된 시점을 중심으로 하여 일정한 추적 윈도우 영역을 설정한다. 추적 윈도우는 $t(k)$ 시점에서 설정한 윈도우 내에 포함된 측정데이터를 사용해서 $t(k+1)$ 시점의 목표물 상태를 추정하고 예측한다.

운동 영상에서 움직이는 목표물을 추적하는 2가지 방법은 첫째, 해석하고자 하는 영역이나 물체가 field-of-view에 존재하게 함으로써 카메라를 고정하고 윈도우를 사용하여 추적하는 방법으로 주변이 악조건인 상황이 아니면 배경 보상을 하지 않아도 추적이 가능한 것점이 있으며, 더욱 안정된 추적을 위하여 배경 보상을 하고자 할 경우 현재 프레임을 추정한 범위만큼 반대로 이동시키면 배경을 완전하게 보상할 수도 있다.

둘째는 목표물이 field-of-view의 중심점에 항상 위치하도록 카메라가 목표물을 따라서 이동하며 추적하는 방식으로 구동장치와 관성관 및 전기적인 잡음 등으로 인하여 실제 카메라의 이동변위는 카메라를 이동시키고자 하는 구동변위와 다르게 되므로 배경 보상 등을 통하여 추적 성능을 보상시킬 수 있다. 본 논문에서는 두 방법을 혼합 적용함으로써 추적 윈도우가 윈도우 영역을 벗어났을 경우에 카메라에 부착된 구동장치를 구동시켜 더 넓은 범위에서 추적이 가능하게 하였다.

4. 운동모델

실제로 이동물체의 계측을 행할 경우, 실제 시스템에 완전하게 일치하는 모델을 구축하는 것은 불가능하므로 변동하는 운동상태를 최대한 근사하게 모델을 선택하는 것이 필요하다. 어떠한 형태의 운동도 충분히 미소한 구간으로 자르면 이 구간 내에서는 등속운동을 하고 있다고 가정할 수 있다. 여기에서는 실제 시스템과 추정모델과의 오차를 시스템 노이즈로 고려하고

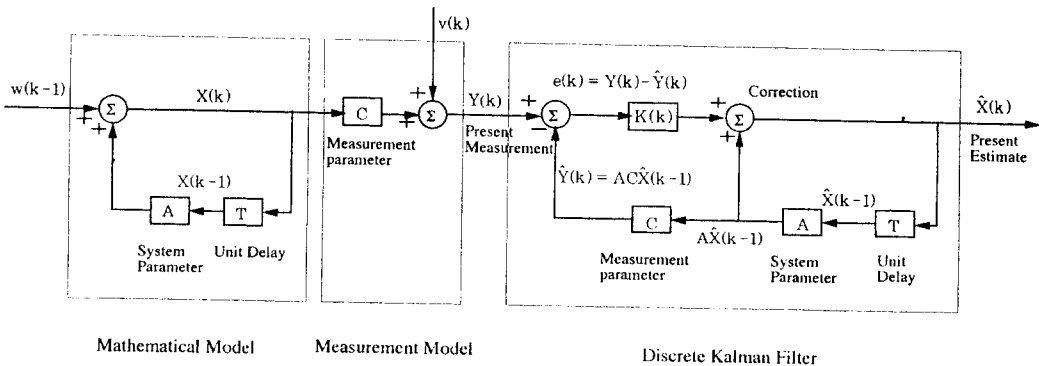


Fig.1 System Modeling and Kalman Filter Structure

칼만필터 알고리즘 추정을 행한다.

등속운동 모델을 사용한 경우 운동방정식은 아래와 같이 표현할 수 있다.

$$\begin{cases} M\dot{x}=0 \\ M\dot{y}=0 \\ M\dot{z}=0 \end{cases} \quad (8)$$

이동물체가 Z방향으로는 일정한 거리를 유지하면서 X, Y방향으로는 등속운동을 한다고 가정하면 위치 및 속도성분을 표시하는 상태변수는 다음과 같이 주어진다.

$$X(t) = [x(t) \quad \dot{x}(t) \quad y(t) \quad \dot{y}(t)] \quad (9)$$

샘플링 시간 Δt 로써 1차 차분근사를 행한 상태변수 X 는 아래와 같이 이산화시켜 시스템방정식을 구할 수 있다.

$$X(k+1) = \begin{bmatrix} \Delta t \dot{x}(k) + x(k) \\ x(k) \\ \Delta t \dot{y}(k) + y(k) \\ y(k) \end{bmatrix} \quad (10)$$

$$= \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ \dot{x}(k) \\ y(k) \\ \dot{y}(k) \end{bmatrix} + w(k)$$

화상위에 관측된 물체의 중심위치를 각각 $x(i)$, $y(i)$ 로 하여 측정방정식을 다음과 같이 구한다.

$$Y(k) = \begin{bmatrix} x(i) \\ y(i) \end{bmatrix} = \begin{bmatrix} f \cdot x(k) \\ z(k) \\ f \cdot y(k) \\ z(k) \end{bmatrix} + v(k) \quad (11)$$

여기서, f 는 초점거리, $x(k)$, $y(k)$, $z(k)$ 는 각 방향의 물체 위치이다.

일반적인 운동모델의 시스템 방정식 및 측정방정식은 다음의 비선형 방정식으로 표시할 수 있다.

$$X(k+1) = F(X(k)) + w(k) \quad (12)$$

$$Y(k) = H(X(k)) + v(k) \quad (13)$$

직각좌표계의 비선형 상태방정식은 선형 시스템으로 표현되더라도 동적 오차가 크게 포함되지 않으므로 어느 정도의 모델링 오차를 감수하여 선형화된 시스템으로 표현할 수 있다. 함수 F , H 를 등속운동 모델로 적용하기 위한 선형화는 테일러급수의 1차항 까지를 취해서 유도한 자코비 행렬을 사용한다. 자코비 행렬 $A(k)$, $C(k)$ 는 다음과 같은 상태 전이행렬로 표현된다.

$$A(k) \cong \left. \frac{\partial F(X)}{\partial X} \right|_{X=X(k)} \quad (14)$$

$$= \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$C(k) \cong \left. \frac{\partial H(X)}{\partial X} \right|_{X=X(k)} \quad (15)$$

$$= \begin{bmatrix} \frac{f}{z(k)} & 0 & 0 & 0 \\ 0 & 0 & \frac{f}{z(k)} & 0 \end{bmatrix}$$

또, 실험에서 설정된 칼만필터의 시스템노이즈, 관측노이즈의 공분산행렬 Q , R 은 다음과 같다.

$$Q = \begin{bmatrix} \sigma_{v1}^2 & 0 & 0 & 0 \\ 0 & \sigma_{v2}^2 & 0 & 0 \\ 0 & 0 & \sigma_{v3}^2 & 0 \\ 0 & 0 & 0 & \sigma_{v4}^2 \end{bmatrix} = \begin{bmatrix} 60 & 0 & 0 & 0 \\ 0 & 90 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 60 \end{bmatrix} \quad (16)$$

$$R = \begin{bmatrix} \sigma_{r1}^2 & 0 \\ 0 & \sigma_{r2}^2 \end{bmatrix} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \quad (17)$$

추정은 칼만게인을 순차적으로 재차 갱신하면서 행한다. 이 칼만필터의 게인은 정상상태의 어느값으로 접근특성을 가진다. 추정과정에 있어서 게인의 산출은 시간적으로 최대한 큰 비율을 점하고 있으며 추정 속도는 이 게인의 계산 과정을 어떻게 하느냐에 따라 다르다. 추정의 고속화로 부터 화상취득 시간내의 추정과정을 여러번 반복하는 것이 가능하다. 결국, 게인 산출 시간을 단축시키는 것으로 부터 보다 세밀한 시각으로 추정 한 값을 얻을 수 있다.

5. 실험장치 및 방법

실험에 사용한 장비는 CCD카메라(Panasonic WV-BL200), 화상처리보드(DT2867-LC), 컴퓨터(386 DX-33MHz), 스텝핑 모터로 구성되고 목표물은 SCARA 로봇 핸드 부분에 부착한 지름 37.6mm의 백색 탁구공을 사용하였다. 화상보드는 NTSC신호를 입력으로 하고 필드단위의 화상을 1/60sec의 샘플링으로 얻을 수 있으며, 해상도는 640x480, gray level은 256(8bit)이다.

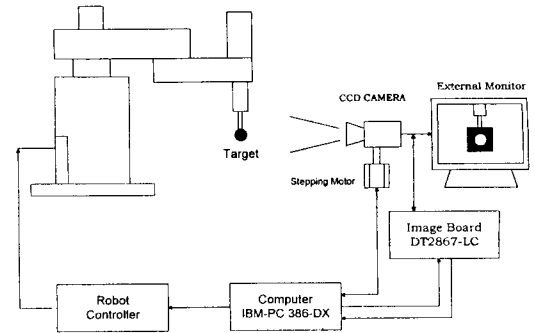


Fig. 3 Overall Configuration Experimental System

Fig.3은 실험장치의 개략도이며 카메라는 목표물 전방 1m 떨어진 곳에서 데이터를 습득한다. 습득한 화상정보는 보드에서 Dual Port Access방식과 입출력 Buffering 방식으로 데이터 흐름의 시간 지연 없이 외부 모니터로 출력과 동시에 컴퓨터와 인터페이스가 이루어진다. 컴퓨터의 입출력카드에는 타이머가 있어 프로그램 실행 시간을 측정하며 물체의 중심위치를 면적 정보로 물체 운동을 고려하고 실시간으로 계측하였다.

Fig.4는 일반적인 화상처리 방법으로써 1 필드 화상 데이터가 1/30sec마다 화상보드를 통하여 컴퓨터에 전송되고 있으며 이때 화상보드로 부터 데이터 전송 종료까지 CPU는 무부하 상태에 있게 된다. 따라서 CPU가 무부하 상태에 있는 이 시간을 이용하여 화상습득과 동시에 알고리즘을 계산하는 병렬처리를 실시하여 실시간 계측을 하였다. Fig.5는 이 방식의 타이밍 선도를 표시한다.

즉, 짝수필드 화상의 습득이 끝난 시점에서 2치화, 중심산출의 전처리를 개시하고 이 처리를 홀수필드 화상의 습득시간내에 종료하여 얻어진 중심위치를 칼만필터에 입력한다. 칼만필터의 추정은 다음의 짝수필드 화상의 습득과 병렬로 실행하고 짝수필드의 화상습득 종료 전까지 이 화상내의 물체위치를 예

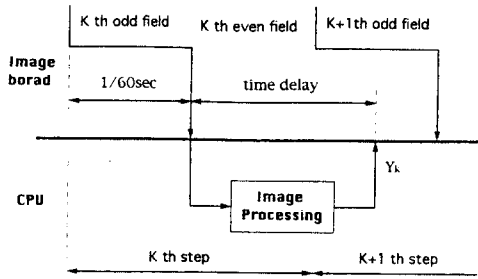


Fig.4 Timing chart using conventional processing

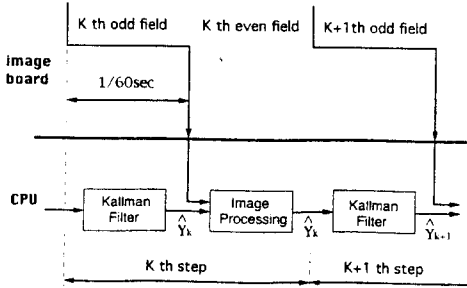


Fig.5 Timing chart using Kalman Filter

측한다. 칼만필터의 추정을 여러번 반복하는 것이 샘플링 시간 내에 가능하면 추정시간을 더 짧게하여 화상습득 샘플링 주기보다 더 짧은 샘플링 주기로 물체의 위치 예측이 가능하다. 이 예측을 전제로 새로이 취한 화상내의 처리영역을 추적원도우로 한정하여 노이즈 경감, 처리시간의 단축을 실현한다.

화상은 256개의 농담치를 가지고 물체를 구분하므로 배경과 목표물이 구분되지 않으면 추적필터가 목표물을 벗어나 발산할 경우가 있으므로 검은색 배경으로 단순화 하였다. 윈도우의 처리영역은 화상내에서 칼만필터의 추정위치를 중심으로 한 가로, 세로 32×32 Pixel을 설정하였다. 컴퓨터에서 로봇트를 구동시킴과 동시에 카메라는 화상데이터를 입력받기 시작하면서 목표물의 중심점을 추적하기 시작한다. Fig.6은 실험에 사용한 추적프로그램의 흐름도이다.

6. 실험 결과

실험은 2차원 상에서 x, y방향으로 목표물의 속도를 변경해가며 직선 운동하는 실험과 sine파와 삼각파 형상의 추적성능을 실험하였다. 각각의 속도 변화에 따른 위치오차를 RMS 오차값과 최대오차값으로 추적성능을 비교하였다. Table.1과 Table.2에 이러한 오차 분포값을 도표화 하였다. 화상의 1 pixel은 1m 떨어진 위치에서 목표물 좌표의 0.8mm에 해당하는 값을 가진다.

Fig.7과 Fig.8은 x, y방향으로 직선운동시 실제 추적값과 추적 실험값을 비교하였고, Fig.9와 Fig.10은 sine파와 삼각파 형상의 추적결과이다. Table.1에서 속도가 증가하면 오차가 커지고, 목표물 지름의 10%이내로 오차가 수렴 시간은 1 sec로 나타났다. 운동체의 속력이 빠르면 최대오차가 큰 값을 보였고, 화상에서 최대오차는 8 pixel로 나타났다.

스텝모터를 구동기로 사용하여 이동물체의 추적시 모터의 미소속도 변화의 제한으로 예측오차를 원만히 감소시키는 수렴성

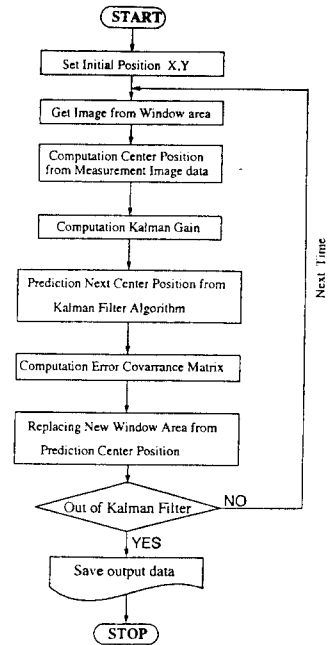


Fig.6 Flowchart of the Tracking Program

에는 한계점이 있었다. 또한, 로봇트가 정지하는 부근에서 오차가 제로 부근에서 머무르지 않고 마이너스 값으로 치우치는 현상이 일어나는데 이것은 로봇트가 감속하여 정지하는 구간에서 로봇트의 빠른 속도에 비해 샘플링 시간이 길기 때문에 일어나는 현상으로 보여진다.

본 실험에서는 비디오 신호 규격에 의해서 샘플링주기를 1/30sec이상 짧게 할 수 없으므로 운동속도가 큰 물체에 대해서는 추종성이 문제가 된다. 여기에 대해서 추정결과와 영향을 조사하기 위해서 계측 정도를 검증하였다. 실험에서 사용한 알고리즘으로 칼만개인을 산출하고 중심점을 예측하는 루틴을 실행하는데 4msec가 소요되었다. 이것은 본 논문에서 제시한 칼만필터의 추정과 화상의 습득을 병렬로 처리하는데 충분한 시간을 알 수 있다.

Table.1 X-direction Error (Sampling Time : 1/30sec)

Velocity	Error	RMS Error (mm)	Maximum Error (mm)
52 mm/sec		1.2	2.9
104 mm/sec		2.3	5.4
198 mm/sec		4.5	10.9
254 mm/sec		5.2	12.3

Table.2 Y-direction Error (Sampling Time : 1/30sec)

Velocity	Error	RMS Error (mm)	Maximum Error (mm)
17 mm/sec		0.2	0.6
32 mm/sec		0.5	1.2
44 mm/sec		1.2	1.8
63 mm/sec		1.9	4.1

또한, 이미지 데이터를 습득(32×32pixel)하고 2치화, 중심점을 산출하는데 소요되는 시간은 타이머 보드에 의해서 계산한 결과 13ms가 소요되었다. 이것은 화상데이터에서 하나의 필드를 습득하는데 걸리는 시간이 1/60sec(16.67ms) 임에 비해서 본 논문에서 제시한 알고리즘 계산시간이 13ms임을 볼 때 실시간 추적기의 구성이 가능함을 보여주고 있다.

7. 결 론

이동물체의 실시간 시각추적을 구현하기 위한 기법의 타당성을 검토하였다. 칼만필터 알고리즘의 연산과 동시에 화상정보의 습득을 병렬처리함으로써 본 알고리즘의 연산시간이 4ms, 이미지 데이터를 습득하고 2치화, 중심점을 산출하는데 13ms가 소요되어 퍼스널 컴퓨터를 이용한 화상처리 실시간 추적기를 구현하는데 시간적으로 충분함을 알 수 있다.

스텝모터를 구동기로 사용하여 이동물체의 추적에는 모터의 미소각도 변화의 제한으로 예측오차를 원만히 감소시키는 수렴성에는 한계점이 있었다.

참고문헌

1. J.Wang and W.J.Wilson, "3D Relative Position and Orientation Estimation Using Kalman Filter for Robot Control", IEEE International Conference on Robotics and Automation, 1992.
2. P.K.Allen and B.Yoshimi, "Real-Time Visual Servoing", IEEE International Conference on Robotics and Automation, pp.851-856, 1991.
3. N. Papanikolopoulos and P.K.Khosla, "Vision and Control Techniques for Robotic Visual Tracking", IEEE International Conference on Robotics and Automation, pp.857-864, 1992.
4. Kohtaro OHBA, Shinya KAJIKAWA, and Hikaru INOOKA, "Model-Based Measurement of 2-D Moving Object", ICIP '92 Proceedings of the 2nd Singapore International Conference on Image Processing, Ref.2, pp.376-379, 1992.
5. F.R. Castella and F.G. Dunnebaek, "Analytical results for x-y Kalman tracking filter", IEEE Trans. on Aerospace and Electronic Systems, Vol.AES-10, pp.891-895, July, 1991.
6. Jeff B.Burl, "A Reduced Order Extended Kalman Filter for Sequential Images Containing a Moving Object", IEEE Trans. on Image Processing, Vol.2, No.3, pp.285-295, July, 1993
7. 안상도, "비전시스템을 이용한 이동물체의 실시간 추적", '94 대한기계학회 호남지부 학술강연회 논문집, pp.97-103, 1994.

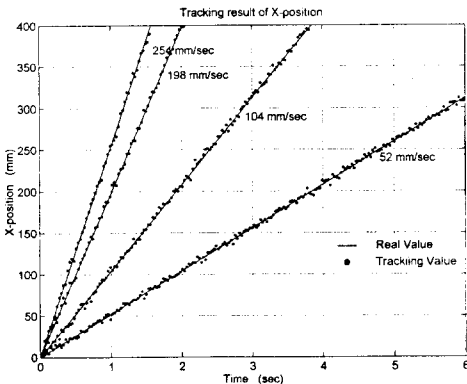


Fig.7 Tracking result of X-position

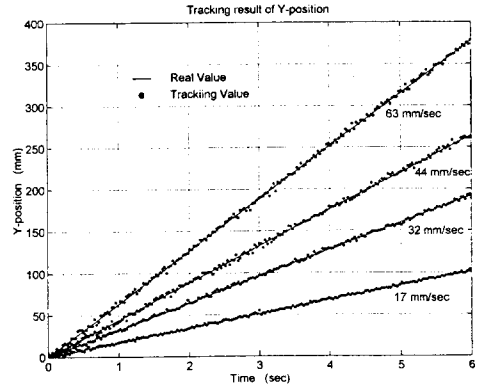


Fig.8 Tracking result of Y-position

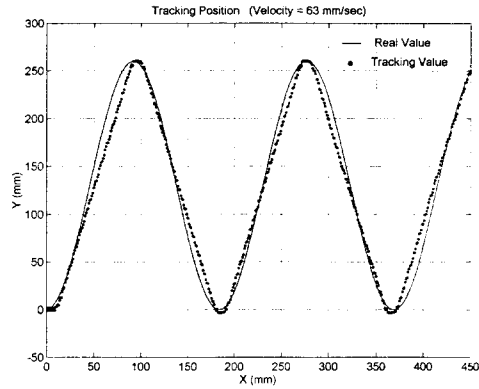


Fig.9 Tracking result of Sine wave

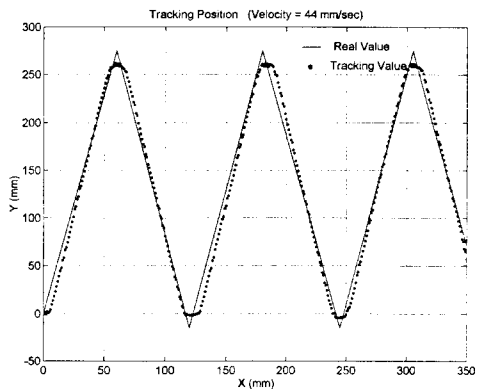


Fig.10 Tracking result of Triangle wave