실시간 자동영상 추적기를 위한 영상영역화의 구현에 관한 연구

°문 종 환* 김 경 수* 김 재 희**

* 한국방송공사 기술연구소　　** 연세대학교 전자공학과 교수

A Study on the Implementation of the Picture segmentation for a Real-Time Automatic Video Tracker System

°Chong Hwan MOON, Kyeong Soo KIM, Jae Hee KIM

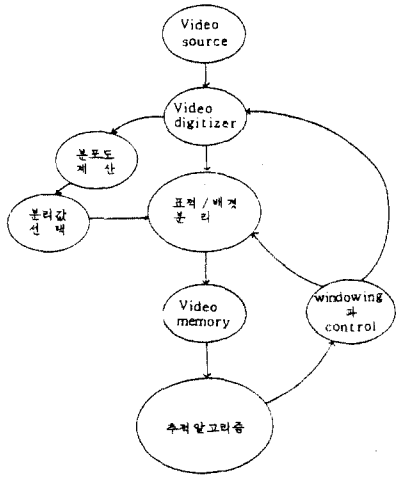
* KBS TRI.　　** Electronic Eng. Dept. of yonsei University

This paper describes a way of implementing the segmentation of 128*128 pixel images to be used as the inputs to a real-time automatic video tracker. The suggested method uses the lowest valley-value of the computed intensity histogram with 16 levels. This method improves smoothing effects and also significantly reduces hardware requirements. Entire segmentation process is caried out in 10msec thus making a real time application possible.

1. 서 론

자동영상추적기(automatic video tracker)란 컴퓨터로 하여금 영상을 처리하여 자동으로 물체를 추적하게 하는 장치를 말한다. 자동영상 추적기는 디지털 기술의 발달로 현재 중심추적(center tracking)등과 같은 간단한 알고리즘을 이용하여 하아드웨어로 실현되어 있다[13]. 자동영상추적기에서의 모든 처리는 실시간 동작을 위하여 매 1/30초마다 모든 계산을 완료되어야 한다.

영상추적기에서 사용하는 영상은 카메라로 부터 얻은 그대로의 영상을 사용하는 방법과 이를 영역화한 후에 이용하는 방법, 2가지로 나눌 수 있다. 현재의 영상을 얻어 추적하는 경우에는 그 계산량이 방대하여 실시간 실현가 현재의 기술로는 어렵다. 그래서, 영상을 일단 영역화한 후에 binary 영상을 사용하게 된다. 이와 같은 방법으로 실시간 동작에 유리한 영상추적기의 기능 블럭선도는 [그림-1]과 같다.



[그림-1] 영상추적기의 기능 블럭선도

영상영역화의 대표적인 방법으로는 1) edge 검출 방법[1,2,3,4] 2) 영역분리 또는 결합 방법[5,6,7] 3) thresholding 방법[8,9,10]으로 나눌 수 있다. thresholding 방법이란 각 영역의 밝기의 차이가 분명할때 이를 밝기를 구별할 수 있는 분리(threshold)값을 정하여 이 값을 기준으로 영역을 분리하는 방법이다. 이 방법은 위의 1), 2)의 방법에 비하여 계산량이 적으므로 실시간을 고려할 때 매우 유리하다.

본 논문에서는 thresholding방법으로 실시간 동작하며, 매 frame마다 자동으로 영역화가 가능한 영상영역화기를 설계하고 그 성능을 실험했다

2. 밝기분포 계산

TV카메라에서 FOV(Field of View)내의 화면을 n * m 행렬 형태로 digitize하면

$$P = (P_{ij})_{n,m} \qquad (1)$$

으로 화소의 밝기 P_{ij} 를 통하여 표시할 수 있다. 이렇게 구성된 행렬 형태를 영역 R이라면, R내에서의 밝기 분포도 $h^R(x)$는 domain [0,d], range [0,r]이 된다. 여기서 d는 A/D 변환기의 dynamic range를 나타내고, r은 영역 R에 포함된 화소의 수이다.

x_i를 x의 domain에서 i번째 element라 하고, x(j)를 영역 R에서 j번째 표본(화소)을 표시한다면 분포도 h (x)는

$$h^R(x_i) = \sum_{j=1}^{r} \delta_{x_i,\, x(j)} \qquad (2)$$

여기서 δ 는 kronecker delta 함수이며

$$\delta_{i,j} = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$$ 이다.

식(2)는 하나의 x에 대하여 미번 영역R내의 표본을 모두 scan하여야 한다. 실제 설계를 위하여 영역 R내의 표본을 빈번한 scan하여 모든 x_i에 대한 본포도를 계산할 수 있는 형태로 표시하면 다음과 같다.

$h^R(x)$를 구성하는 (d+1)개의 1차원 array를 영역 R에서 처리하기 전에 미리 '0'으로 만들어 준다.

$$h^R(xi) = 0 \quad for \quad i = 1, 2, \ldots\ldots, d+1 \qquad (3)$$

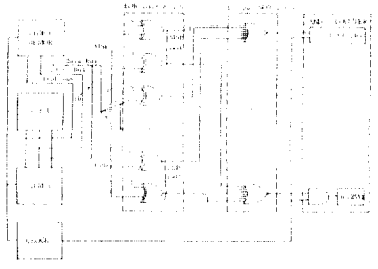
영역 R에서 각 표본(화소)를 처리할 때 마다 (d+1)개의 array중 단지 한나만 증가시킨다.

$$h[x(j)] \longleftarrow h[x(j)] + 1 \qquad (4)$$

전체영역을 scan하고 나면 h는 이 영역내의 화소의 밝기에 대한 본포도를 갖게되며, 이때 구한histogram을 영역 R에서의 본포도라 하고, 다음의 등식을 만족한다.

$$r = \sum_{i=1}^{d+1} h^R(x_i) \qquad (5)$$

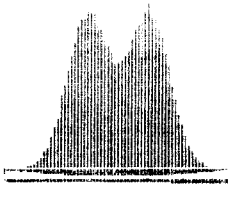
위의 방법을 써서 구성한 본포도 계산 회로의 블럭도는 [그림-2]와 같다.



[그림-2] 밝기 본포도 계산기의 구성도

3. Minimum error thresholding과 본리값 추출

본리값의 선택은 영상의 밝기 본포도가 2개의 비슷한 크기의 peak를 가지며, 이들 peak 사이에서 계곡(vally)이 존재하는 [그림-3]과 같은 bimodal 형태일 경우 비교적 쉽다. 이러한 형태의 본포도에서 배경과 표적을 본리하기 위한 본리값을 정하는 방법에는 1) 전체화면에서 표적이 차지하는 면적의 비율이 알려져 있다면 본포도를 통하여 전체화면에서 이 비율을 줄 수 있는 본리값을 선택하고 [11], 2) 표적과 배경의 밝기영역이 충분히 떨어져 있다면, 두 peak사이의 계곡값 중 최소값을 찾아 본리값으로 선택하는 방법등 크게 2가지로 나눌 수 있다[8].



[그림-3] 밝기 본포도의 예

주어진 본포도에서 표적과 배경에 해당하는 부분 각각을 독립적인 확률본포함수(p.d.f)로 표시하고 밝기본포도를 Gaussian확률본포로 근사화한 후, 표적과 배경 영역으로 나타낸 $P_r(x/T)$, $P_r(x/B)$를 표적과 배경의 평균밝기 m_T, m_B 와 표준편차σ_T, σ_B로 표시하면

$$P_r(x/T) = \frac{1}{2\pi\sigma_T} e^{-\frac{(x-m_T)^2}{2\sigma_T^2}} \qquad (6)$$

$$P_r(x/B) = \frac{1}{2\pi\sigma_B} e^{-\frac{(x-m_B)^2}{2\sigma_B^2}} \qquad (7)$$

이 된다.

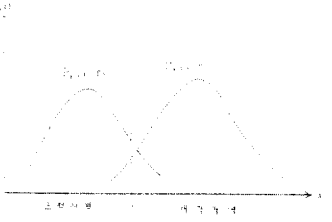
여기서 표적과 배경의 면적을 안다면, Hypothesis testing이론에 의하여 표적을 배경(B/T)으로 배경을 표적(T/B)으로 잘못 판별할 error가 일어날 확률 P_r(error)는

$$P_r(error) = P_r(T/B)P_r(B) + P_r(B/T)P_r(T) \qquad (8)$$

이고, 위의 error확률을 최소로 하는 Bayes decision rule은 다음과 같이 된다.

$$\frac{P_r(x/T)}{P_r(x/B)} \underset{B}{\overset{T}{\gtrless}} \frac{P_r(B)}{P_r(T)} \qquad (9)$$

위 식에서 P (x/T)와 P (x/B)는 조건부 확률을 나타 낸다.[그림-4a]는 이러한 확률에 대한 예이다.



[그림-4a] 조건부 확률의 예

식(8)에 COST를 부가하고, 일반화한 P_r(error)를 최소로하는 Bayes decision rule은

$$\frac{P_r(x/T)}{P_r(x/B)} \underset{B}{\overset{T}{\gtrless}} \frac{C(T/B) - C(B/B)}{C(B/T) - C(T/T)} \cdot \frac{P_r(B)}{P_r(T)} \qquad (10)$$

이다. 이식을 등식으로 하여 풀면 표적과 배경을 본리하는 본리값 t를 구할 수 있다.

식(10)에서 COST C(B/B), C(T/T)를 '0'으로, 잘못 판별한 경우에 대한 COST C(B/T)와 C(T/B)를 같게하고, 전체 화면에서 표적이 차지하는 비율 θ를

$$\theta = \frac{P_r(T)}{P_r(B) + P_r(T)} \qquad (11)$$

로하여 등식으로 풀은 후, x 대신 t를 대입하면

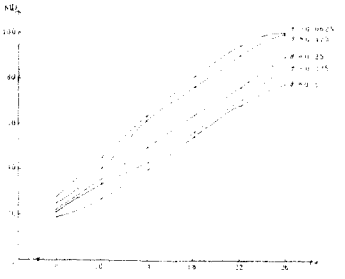
$$\frac{P_r(t/T)}{P_r(t/B)} = \frac{1-\theta}{\theta} \qquad (12)$$

이다. 여기에 식(6), (7)을 대입하고, $\sigma_T = \sigma_B = \sigma$ 로 표적과 배경의 분산 정도가 같다고 하면, 본리값t 는

$$t = \frac{m_T + m_B}{2} + \frac{\sigma^2}{m_B - m_T} \ell n \frac{\theta}{1-\theta} \qquad (13)$$

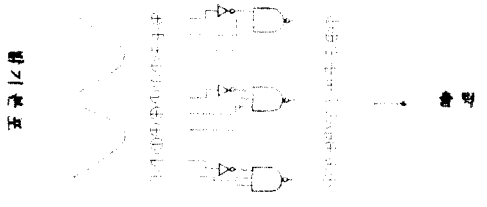
으로 나타난다. 이것이 사전확률을 아는 것을 가정으로 최소error를 주는 분리값을 구하는 식이지만 본 논문에서 취급하는 영상의 내용상 사전확률을 아는 것은 불가능하므로 이것으로 분리값을 결정할 수는 없다. 그러나, 어떤 분리값이 평균 밝기, 표준편차, 표적의 전체화면에 대한 비율 등 3가지 화면의 성질에 의하여 조건부로 정하여 짐을 알 수 있다.

[그림-4b]는 화면의 성질을 넣어 컴퓨터 시뮬레이션에 의하여 작성된 실제 영역화가 가능한가를 나타내는 표이다. 화면의 성질로는 배경과 표적의 평균밝기의 차 MD(mean difference), 표준편차, 그리고 표적의 비율 θ, 3개이다. [그림-4b]는 thresholding 방법으로 영상영역화가 가능한 화면의 성질에 대한 한계치를 정할 수 있음을 보이고 있다.



[그림-4b] thresholding방법으로 영상영역화가 가능한 화면의 성질에 대한 그래프

위에서 말한 hresholding에의해 영상영역화가 가능한 화면을 최저계곡값으로 영역화했을 때, 두 peak사이의 간격과 표준편차라는 조건에 의하여 최소 error를 반드시 얻을 수는 없지만, 사전확률을 알 수 없고 실시간 처리에 유리하므로 최저 계곡값으로 분리값을 취하는 방법이 쓰인다.



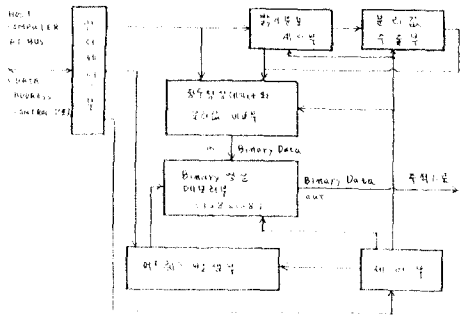
[그림-5] 계곡값중 최소치를 찾는 원리

영역화가 가능한 화면의 밝기분포도는 [그림-3]과 같은 bimodal한 특성이므로 계곡 바로 전의 기울기와 지난후의 것의 부호를 비교함으로써 [그림-5]와 같은 원리를 사용한 계곡값중 최소치를 매 frame 마다 자동으로 찾아내는 분리값 추출기를 설계했다.

4. 실험과 결과 고찰

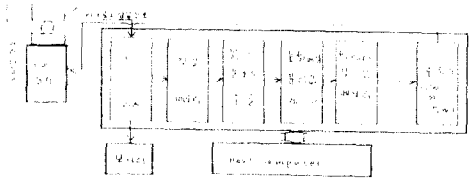
(1) 영상영역화기와 실험 시스템의 구성

영상영역화기는 위에서 설계한 1) 밝기분포 계산부 2)분리값 추출부와 실험과정에서 언급될 3) 표적/배경 분리및 영상memory부등 크게 3과정으로 구성된다.



[그림-6] 영상영역화기의 구성도

영상 data, 제어신호, address 등은 pc AT bus를 이용하였다. 데이타 입력으로 사용될 영상은 512*512 pixel의 영상을 128*128로 축소(그림-8)한 후에 사용했다. [그림-6]이 영역화기의 구성도이며, [그림-7]은 영역화기의 성능을 실험하기위한 시스템의 구성도이나 여기에서는 실험의 목적에 맞도록 재편했다.



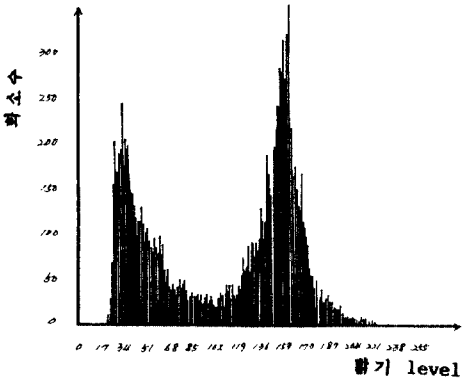
[그림-7] 실험 시스템의 구성

(2) 밝기분포 계산 실험

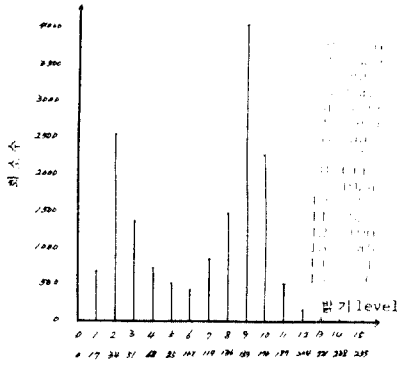
밝기분포도는 명암 레벨에 따른 화소수를 세어서 X축에 명암의 레벨을 Y축에 화소의 수를 표시함으로써 얻어진다. 8bit의 128*128 영상의 밝기 분포도는 소프트웨어 처리에 의하여 [그림-9]와 같이 얻어진다

[그림-8] 영상data 입력으로 사용된 128*128 영상

실제회로에서는 본포도의 smoothing효과를 얻기 위하여 상위 4bit만으로 구성된 화소의 명암 레벨 16개중 하나가 되게하여 카운트하도록 했다. [그림-8]과같은 영상 데이타를 본포도 계산부의 입력으로 했을때, 그



[그림-9] 128*128 영상의 256레벨 밝기 본포도



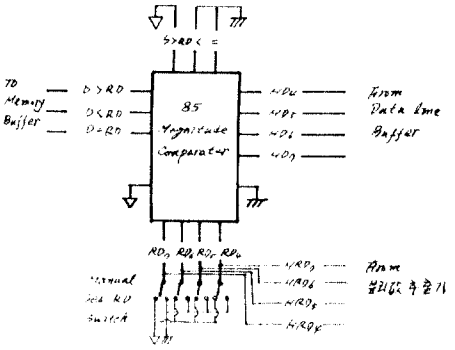
[그림-10] 128*128 영상의 16 level 밝기본포도

결과는 [그림-10]과 같다. 이것은 [그림-9]와 같은 원래의 본포도와 비교할때 원하던 바의 smoothing 효과를 달성하고 있음이 확실하고, 앞에서 설계한 본리값 추출기의 동작이 가능케 될것임을 짐작케한다.

(3) 영역화 실험

밝기본포도에서 구한 16개 level의 화소수를 서로 이웃한 것끼리, 낮은 level에서 높은 level 쪽으로 비교해가면 16개의 '+' 또는 '-'부호를 얻을 수 있다. 이 16개의 레벨이 앞에서 말한 최저 계곡값 선택기법에 의해 인버터와 NAND 게이트로 이루어진 본리값 판별 회로에 입력되면 그중 한 level을 출력하여, [그림-11]과 같은 binary data화부의 본리값을 설정한다. 여기에서 한후레임의 영상 data는 '0' 또는'1'의 레벨로 본류되어 영상메모리에 저장하면 영역화는 완료된다.

단계별로 영역화를 시험하기 위하여 binary data화부의 기준값을 '0111'로 설정한 후, [표-1]의 데이타



[그림-11] Binary data화부의 interface

[표-1] 영역화시험 입력으로 사용된 data

9000 :	3	7	8	3	13	17	18	17	23	27	28	23	37	38	3F
9001 :	43	47	48	4F	53	57	58	5F	63	67	68	6F	73	77	78	7F
9002 :	83	87	88	8F	93	97	98	9F	A3	A7	A8	AF	B3	B7	B8	BF
9003 :	C3	C7	CB	CF	D3	D7	DB	DF	E3	E7	EB	EF	F3	F7	FB	FF

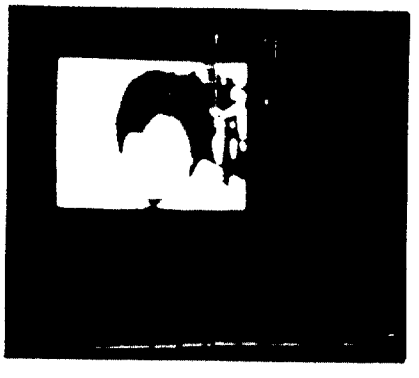
[표-2a] 상위 4bit '0111'로 영역화한 결과

9000 :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9001 :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9002 :	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
9003 :	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

[표-2b] 상위 4bit '1011'로 영역화한 결과

를 영역화한 결과는 [표-2a]와 같이,'1011'로 설정했을 때는 [표-2b]와 같이 반전되어 나타났다. 다음에는 기준값 설정을 auto로한 후, [그림-8]의 인물영상 카메라로 잡아 영역화한 결과 [그림-12]와 같다.
이것은 다시 기준값을 [그림-10]에서 얻은 최저 계곡 level '6'(밝기 102)으로 설정했을 때의 영역화 화면 차이가 없었다.

이것으로 실제 영상data의 영역화에서 본리값 추출, data의 본리/저장 등이 이루어짐을 알 수 있다.



[그림-12] 영역화한 128*128 화소의 영상

(4) 실시간 동작

실시간 동작을 고려할 때, RAM의 억세스타임에 큰

게 그 동작이 좌우된다. Fast 알고리즘[12] 적용시 32*32의 부영상을28 개 읽어 내는데드는 시간 이내에는 모든 계산을 끝낸다는 가정에서 RAM의 억세스타임 t를 계산해 보면

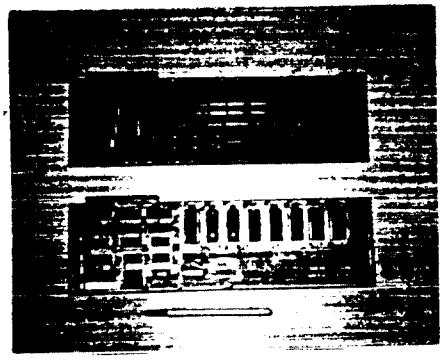
$$t\ \{(128 * 128) + (28 * 32 * 32)\} < 1/30 \text{ 초}$$

$$t < 740 \text{ nsec} \quad (13)$$

식(13)에 의하면 RAM의 억세스타임이 740 nsec이하가 되어야 한다는 조건이 된다. 실제 RAM의 억세스타임을 데이타의 흐름을 따라 계산해 보면

$$\{(30*2) + 500\} = 560 \text{ nsec} \quad (14)$$

로서, 실시간 동작함을 알 수 있다. 여기서 60 nsec는 버퍼 등에서, 500 nsec는 clock으로 8 MHz를 사용했을 때 RAM을 억세스 하는데 소요되는 시간이다.



[그림-13] H/W로 구현한 실시간 영역화기 본비값 추출부(위), 영역화 메모리부(아래)

5. 결 론

실시간 자동영상 추적기의 입력에 128*128 화소의 영역화 영상을 제공하기 위하여 밝기본로 계산, 본비값 추출 등 중요한 부분을 설계하고, data의 비교 등 필요한 모든 신호와 계산을 카운터, 비교기, NAND, 인버터 등 TTL 소자를 이용한 하아드웨어로 구성 실험하였다.

실제 실험에 의하여 (1) 본로도 계산이 영상data의 상위 4bit 만으로도 가능하며 smoothing 효과를 더해주고, 하아드웨어를 1/16로 줄일 수 있게 했다.

(2) 본로도의 smoothing을 이용한 하아드웨어 기법으로 본비값 추출기의 동작이 확실히 됨을 보였다.

(3) 추정한 RAM의 억세스 타임 :740nsec에 비하여 실제 560 nsec로서 실시간 동작함을 알 수 있다.

(4) 전체적으로 매 frame 마다 자동으로 실시간 영역화가 됨을 보였다.

영상추적, 영상검색 등에서 가장 기본적으로 이루어져야할 영상영역화의 실시간 동작을 구현했다. 더욱 error확률을 작게할 수 있는 실시간 영역화 기법이 연구 보완되어야 한다.

참 고 문 헌

1. W. Frei & C.C.Chen, "Fast Boundary Detection: A Generalization and A New Algorithm," IEEE. Trans. Comput. vol.C-26, no.10, Oct. 1977.pp. 988-998
2. C.J.Jacobus & R.T Chien, "Two New Edge Detectors," IEEE Trans. PAMI, vol.PAMI-3,no.5,Sep 1981, pp.581-592
3. W.K.Pratt, Digital Image Processing, Wiley-Interscience, New York, 1978.
4. D.H.Ballard & C.M.Brown, Computer vision, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1982
5 P.M.Narendra & M.Goldberg,"Image Segmentation with Directed Trees, "IEEE Trans. PAMI, vol. PAMI-2, no.2, Mar.1980, pp.185-191
6. S.L.Horowitz & T.Pavlidis, " Picture Segmentation by a Tree Traversal Algorithm," J.Ass.Comput.Mads.,vol.23,1976. pp.368-388
7. S.L.Horowitz & T.Pavlidis, "Picture Segmentation by a Directed Split-and-Merge Procedure, "Proc.2nd Int.Joint. Pattern Recognition, Aug. 13-15,1974, pp.424-433
8 A.Rosenfeld & A.C.Kak, Digital Picture Processing, Academic Pres, N.Y.1976.
9. J.S.Weszka, R.N.Nagel, and A.Rosenfeld, " A Threshold Selection Technique," IEEE Trans Comput.vol.C-23,no.12,Dec.1974,pp.1322-1326
10. S.W.Zucker, A.Rosenfeld and L.S.Davis,"Picture Segmentation by Texture Discrimination ,"IEEE Trans.Comput.vol.C-24,no.12,Dec.1975 , pp.1228-1233
11. H.L.Van Trees, Detecton, Estimation, and Modulation Theory, John wiley and Sons, Inc. , New York, 1968
12. 김정수, 송유섭, 이상욱, " Fast 2-D Moving Target Tracking Algorithm, " 전자공학회지, 제22권 제1호, 1월, 1985년
13. Alton L. Gilbert et al.," A Real-Time Video Tracking System" IEEE Trans. PAMI,vol.PAMI-2, Jan.1980,pp.47-56