

動映像에서 얼굴의 눈위치
追跡 및豫測 시스템 開發

성균관대학교 일반대학원
정보공학과 석사4기
9675978 정윤경

목차

1. 서론

2. 기존 연구 고찰

2.1 얼굴 인식 관련연구

2.2 동영상에서의 물체 움직임 특성

2.3 움직임 추정 (motion estimation)

2.4 신경망 관련연구

3. 제안한 시스템

3.1 연속적인 얼굴영상에서 눈위치 탐지

3.2 신경망을 이용한 눈위치 예측 모델

4. 실험 및 결과

5. 결론

1. 서론

응용 : 카메라의 제어기법, 압축기법, 로보트의 상대방 움직임 예측.

장점 : on-line으로 입력된 데이터의 unsupervised 학습이 가능하다.

정보사회로 발전하면서 우리가 접해야 하는 자료의 양도 기하급수적으로 증가함에 따라 이러한 자료를 처리하는 여러가지 방법이 개발되었다. 본 연구에서 다루려고 하는 기법은 그중 시계열 데이터 예측에 관련된 것이다.

시계열 데이터의 분석 및 예측은 금융, 증권, 환율, 상품의 손익 계산, 이자율 예측 등의 경제관련 분야와 교통량 예측, 기계의 오류 검사, 기상 예측, 여객기의 탑승객 수 예측, 전기 수요 예측, 최적 set-up 예측, 작업 흐름 시간의 예측, 동적 팔 운동 균력 예측, 음성 인식, 잡음 예측, 통신망에서의 traffic prediction, 한국어 단어 범주 예측, GNP 예측, 실업자 비율 예측 등의 공학분야에 많이 활용되어 왔다. 본 연구에서는 이러한 시계열 데이터를 사람의 움직임에 적용하여 자동 카메라의 제어기법이나 자율적 로봇의 상대방 움직임 예측 등에 활용하도록 한다.

예측하는 방법에는 일반적인 통계적 방법과 유전자 알고리즘 및 신경망을 이용하는 방법이 있다.

시계열 데이터의 접근 방법으로는 미래의 시계열 데이터에 대한 예측과 dynamic modelling이 있다. 시계열 데이터 예측은 일정 기간의 데이터 상태를 입력하여 다음 시점의 데이터를 예측하고 오류를 측정하는 것이고 dynamic modelling은 주어진 상태 공간을 예측자에게 입력하여 얻어진 시계열과 실제 시계열 데이터가 합치되도록 하는 것이다.

2. 기존 연구 고찰

예측에 쓰이는 데이터의 종류는 다양하다. 그 중에는 laser data, synthetic data, 환율, 음악, 잠자는 사람의 생리학적 자료, 소행성의 천체 물리학 자료 등이 있다. 본 연구에서 다루는 데이터는 사람의 움직임을 나타내는 속도와 방향 벡터이다.

NMSE(normalized mean squared error)

$$NMSE(N) = \frac{\sum_{k \in T} (observation_k - prediction_k)^2}{\sum_{k \in T} (observation_k - mean_k)^2} \approx \frac{1}{\sigma_T^2} \frac{1}{N} \sum_{k \in T} (x_k - \hat{x}_k)^2$$

여기서 $k=1 \dots N$ 으로 훈련 집합 T 가 보유한 패턴을 열거하는 것이고 $mean_T$ 와 $\hat{\sigma}_T^2$ 은 T 의 목표값에 대한 평균과 편차라고 정의했을 때, 데이터에 대한 기존의 예측 신경망의 성능은 다음과 같다.

laser data

	method		computer	time	NMSE (100)	natL
W	conn	1-12-12-1;lag 25,5,5	SPARC2	12hrs	0.028	3.5
Sa	loc lin	low-paa embed, 8dim, 4nn	DEC3100	20min	0.080	4.8
McL	conn	feedforward, 200-100-1	CRAY Y-MP	3hrs	0.77	5.5
N	conn	feedforward, 50-20-1	SPARC1	3weeks	1.0	6.1
K	visual	look for similar stretches	SG Iris	10sec	1.5	6.2
L	visual				0.45	6.2
M	conn	feedforward, 50-350-50-50	386PC	5days	0.38	6.4
Can	conn	recurrent, 4-4c-1	VAX8530	1hr	1.4	7.2
U	tree	k-d tree;AIC	VAX6420	20min	0.62	7.3
A	loc lin	21dim, 30 nearest neigbh's	SPARC2	1min	0.71	10.
P	loc lin	3 dim time delay	Sun	10min	1.3	
Sw	conn	feedforward	SPARC2	20hrs	1.5	
Y	conn	feedforward, weight-decay	SPARC1	30min	1.5	
Car	linear	Wiener filter, width 100	MIPS3230	30min	1.9	

Money, Music, Stars, Heart

고차원 합성 데이터(high-dimensional synthetic data)

	method		computer	time	NMSE(15)	NMSE(30)
ZH	conn	...-30-30-1 and 30-100-5	CM-2(16k)	8days	0.086	0.57
U	tree	k-d tree;AIC	VAX 6420	30min	1.3	1.4
C	conn	recurrent, 4-4c-1	VAX 8530	n/a	6.4	3.2
W	conn	1-30-30-1;lags 20,5,5	SPARC2	1day	7.1	3.4
Z	linear	36AR(8), last 4k points	SPARC	10min	4.8	5.0
S	conn	feedforward	SPARC2	20hrs	17.	9.5

<참고문헌>

<JOJY95> Jose C. Principe, Jyh-Ming Kuo, "Dynamic Modelling of Chaotic Time Series with Neural Networks", NIPS, 1995

BP(TDNN8-14-1), no momentum, step size of 0.001

500 iteration training,

MSE : 0.000288

TDNNGF

trajectory length 14,

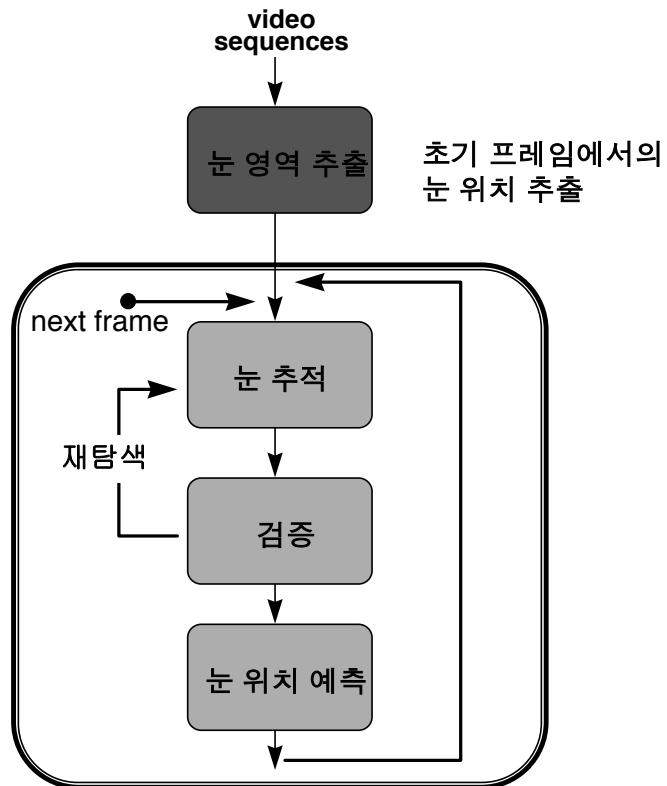
같은 signal에 대해 500번 반복.

MSE : 0.000648

2.2 신경망 관련연구

3. 제안한 시스템

3.1 연속적인 얼굴영상에서 눈위치 탐지 모델



3.2 히스토램 매칭에 기반한 눈영역 추출

실시간 환경에서 회의자의 위치를 파악하기 위해서는 얼굴 영역에서 눈의 위치를 추출하는 알고리즘이 단순해야 한다. 그러나 알고리즘이 단순해지면 눈동자 추출의 정확성이 떨어지는 문제점이 발생하기 때문에 실시간화에서 눈의 영역을 찾는 연구 결과는 정확성과 속도 모두를 만족하지 못하였다. 본 연구에서는 히스토그램 매칭 기법을 제안하여 처리 속도를 빠르게 하고 동시에 정확성을 보장하는 좋은 결과를 얻을 수 있었다.

3.2.1 히스토그램 매칭 기법

가. 히스토그램 매칭 기법의 개요

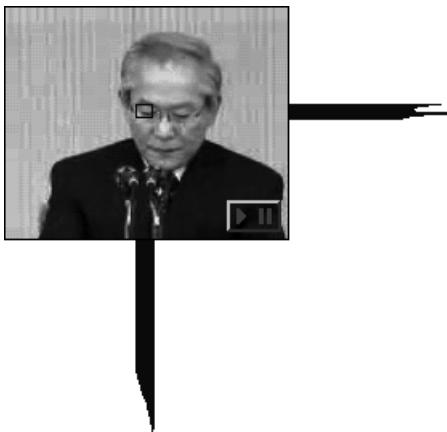
히스토그램 매칭 기법은 매칭하고자 하는 영상의 부위를 수직/수평 히스토그램값으로 변환하여 템플릿화 함으로써 정합을 수행하고자 하는 방법이다. 히스토그램 매칭법을 사용하기 위해서는 정합하고자 하는 영상의 히스토그램이 다른 영상 부위의 히스토그램과 뚜렷이 구별되는 특징을 가지고 있어야 한다. 히스토그램 템플릿 구축시 양쪽 눈이 아닌 한쪽 눈만을 가지고 구성해서 실험한 결과 질이 좋은 영상에 대해서는 좋은 결과를 보였으나, 정면에서 측면으로 바뀌는 등 영상의 변화가 심한 경우에는 히스토그램의 변화치가 상대적으로 크기 때문에 잘못된 부위를 눈으로 오인식 하며, 피실험 대상자의 눈이 작은 경우에도 마찬가지로 주위의 작은 점들을 눈으로 오인식하는 경우가 생겼다. 따라서, 본 실험에서는 얼굴의 방향에 구애받지 않도록 양쪽 눈을 템플릿으로 설정하였다.



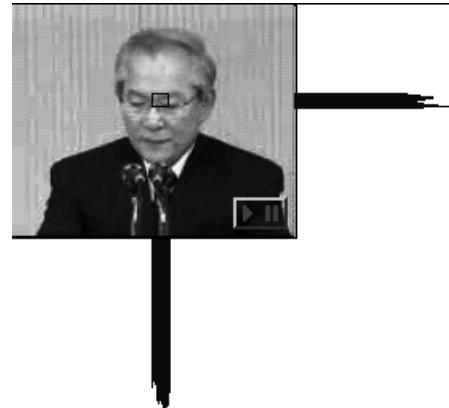
(a-1) frame 1



(a-2) frame 6



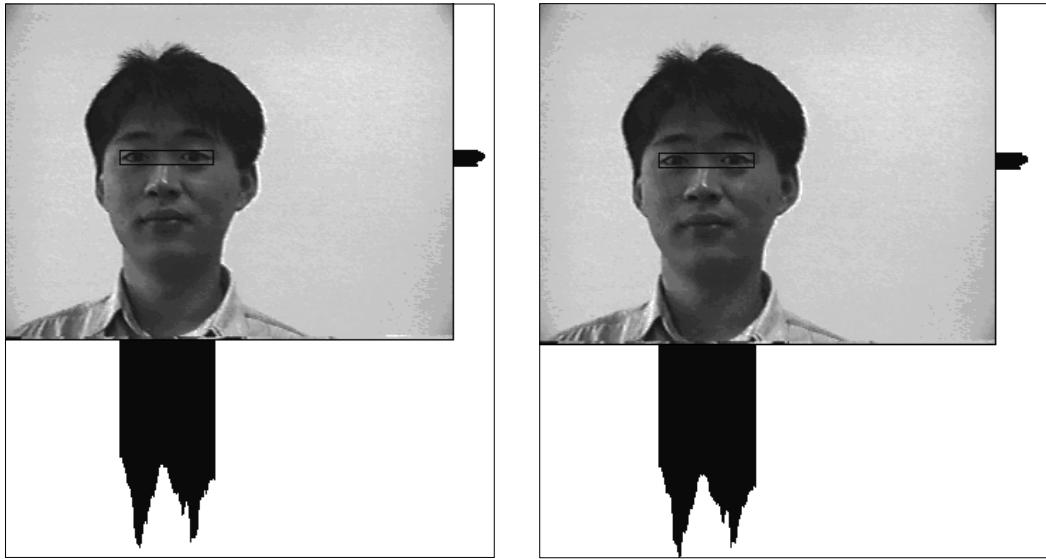
(b-1) frame 3



(b-2) frame 4

그림 4.6 히스토그램 매칭기법을 한쪽 눈에 적용

그림 4.6은 일반적인 얼굴 영상에서 눈 영역의 색상을 x와 y축을 기준으로 누적하였을 때의 히스토그램 그래프이다. 대략적으로 눈의 영역을 지정하여 히스토그램의 가로값으로 판별할 때 눈과 눈 사이의 간격은 최소치를 가지고 눈동자가 있는 부분은 최대치를 갖게 된다. 마찬가지로 히스토그램의 세로값은 눈동자가 있는 부분에서 최대치를 가진다. 히스토그램의 가로값과 세로값을 가지는 배열을 설정하여 히스토그램을 템플릿화하여 저장하고 적절한 매칭 함수를 사용하여 얻어진 다음 프레임에서의 눈영역 추출은 다음과 같다.



(a) frame 1

(b) frame 2

4.7 양쪽 눈 영역의 히스토그램

위의 결과와 같이 현재 프레임의 히스토그램이 이전 프레임과 매우 유사하여 최적화된 매칭 영역이 눈의 영역과 일치함을 알 수 있다.

나. 히스토그램 매칭 알고리즘

매칭 알고리즘에 쓰인 함수는 다음과 같다.

이전 프레임에서 얻어진 눈의 영역을 $P_{eye}(P_{sx}, P_{sy}, P_{ex}, P_{ey})$ 이라 정의 한다.

$(P_{sx}, P_{sy}, P_{ex}, P_{ey})$ 은 얻어진 왼쪽과 오른쪽 눈의 영역을 포함하는 최소 사각형의 시작과 끝점의 (x, y) 좌표이다. 앞으로의 계산을 위해 얻어진 눈 영역의 넓이와 높이를 다음과 같은 변수로 정의 한다.

$$width = (P_{ex} - P_{sx})$$

$$height = (P_{ey} - P_{sy})$$

히스토그램을 템플릿화한 Tx_i, Ty_j 의 값은 다음과 같이 구해진다.

$$Tx_i = \sum_{y=P_{sy}}^{P_{ey}} I(P_{sx}+i, iy)$$

$$Ty_j = \sum_{x=P_{sx}}^{P_{ex}} I(ix, P_{sy}+j)$$

이렇게 이전 프레임에서 얻어진 히스토그램 템플릿으로 다음 프레임에서의 눈의 위치를 추적한다. 우선, 탐색 구간을 ($S_{sx} \leq x \leq S_{ex}$, $S_{sy} \leq y \leq S_{ey}$)로 정한다. 본 실험에서 탐색구간은 이전 프레임에서 얻어진 $width$ 의 $1/2$ 값과 $height$ 의 길이를 이용하여 다음과 같이 임의로 지정하였다. 눈 영역 추출의 실행 시간은 탐색구간의 넓이에 영향을 받는다.

$$S_{sx} = P_{sx} - \frac{width}{2}, \quad S_{ex} = P_{ex} + \frac{width}{2},$$

$$S_{sy} = P_{sy} - height, \quad S_{ey} = P_{ey} + height$$

탐색구간내에서 현재 히스토그램을 취득하는 좌표를 (x, y) 라 하고 이 때의 히스토그램 값 Hx_i, Hy_j 는 다음과 같이 구한다.

$$Hx_i = \sum_{iy=y}^{y+width} I(x+i, iy)$$

$$Hy_j = \sum_{ix=x}^{x+height} I(ix, y+j)$$

위의 식으로 구해진 Tx_i, Ty_j, Hx_i, Hy_j 의 값을 다음과 같은 매칭 유사 판별 함수 $E_{(x,y)}$ 에 대입하여 탐색구간 ($S_{sx} \leq x \leq S_{ex}$, $S_{sy} \leq y \leq S_{ey}$) 안에서의 최소값을 구하고 이 때의 (x, y) 좌표를 얻어낸다.

$$E_{(x,y)} = \sum_{i=0}^m (Tx_i - Hx_i)^2 + \sum_{j=0}^n |Ty_j - Hy_j|$$

$$\min(E_{(x,y)}) \quad (\text{단}, \quad S_{sx} \leq x \leq S_{ex}, \quad S_{sy} \leq y \leq S_{ey})$$

얻어진 시작점 좌표 (x, y) 에서 눈 영역의 높이와 넓이를 더하여 끝점을 계산한 사각형 좌표 $(x, y, x+width, y+height)$ 가 최종적으로 추출된 현재 프레임에서의 눈의 영역이 된다.

다. 히스토그램 매칭 결과

그림 4.8은 캠코더를 통해 얻어진 실제 영상이다. 템플릿의 크기가 정적이기 때문에 눈의 영역 전체를 추출하지는 못하였으나, 실험 영상이 급격한 속도로 전진하고 있음에도 정확히 눈의 중심 영역을 찾아내고 있음을 알 수 있다.



frame 1



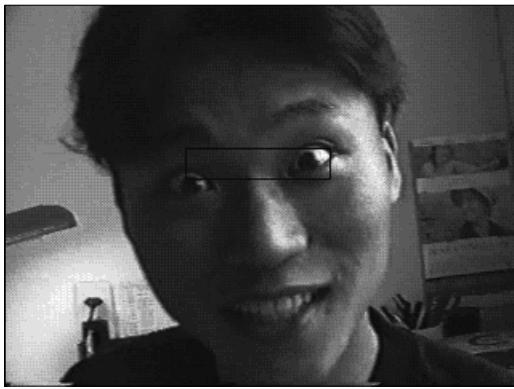
frame 3



frame 5



frame 7



frame 9

그림 4.8 전진하는 영상의 경우

그림 4.9는 인터넷에서 생중계되고 있는 TV 동영상을 취득하여 실험 영상으로 사용한 예이다. 다음 영상에서는 카메라맨이 이동하는 관계로 배경화면이 급격하게 변화였으나 좋은 결과를 보였다.



frame 1



frame 2



frame 3



frame 4



frame 5



frame 6

4.9 배경이 복잡한 영상의 경우

그림 4.10은 인터넷에서 취득된 데이터로 정면 얼굴에서 약간 옆면 얼굴로 바뀌는 경우나 눈이 아래로 깔리면서 눈동자가 보이지 않는 경우에도 좋은 결과를 보였다.



frame 1



frame 2



frame 3

(a)



frame 1



frame 2



frame 3



frame 4

frame 5

frame 6

(b)

4.10 얼굴 방향의 변화가 큰 경우

그림 4.11은 조명의 급격한 변화에도 불구하고 좋은 결과를 보인 경우의 영상이다.



이전 프레임

현재 프레임

4.11 조명의 변화가 큰 경우

그림 4.12는 실패한 결과의 예이다. 입력 영상은 인터넷에서 취득하였다. frame 5의 앞면 고개숙인 영상에서 고개를 들고 있는 옆면 영상으로 바뀌면서 급격한 변화때문에 템플릿 적용이 제대로 되지 못한 경우이다.



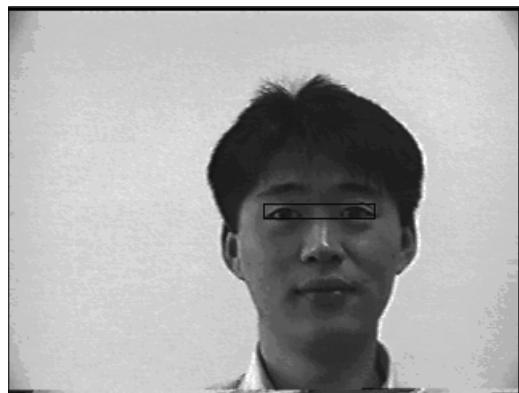
frame 1

frame 2

frame 3

4.12 얼굴 방향의 변화가 심한 경우

그림 4.13은 프레임 간의 간격을 매우 크게 하여 움직임이 급격히 변하도록 조작한 데이터를 입력했을 때의 결과이다. 이 때, frame 5에서는 눈의 위치가 이전 프레임에서 정해진 탐색구간안에 위치하지 않기 때문에 다른 곳이 선택되어졌다.



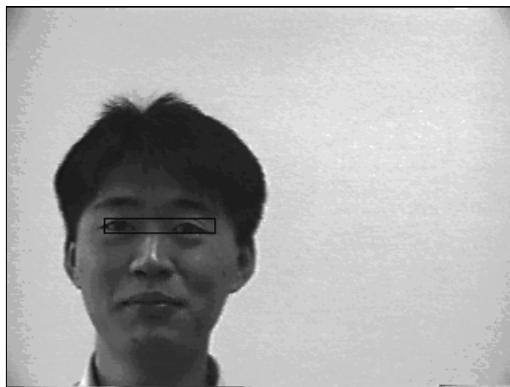
frame 1



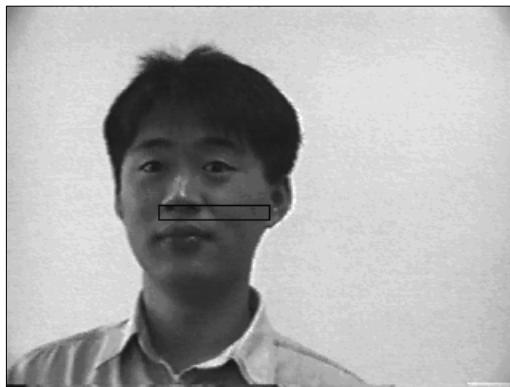
frame 2



frame 3



frame 4



frame 5

4.13 이동이 급격한 경우

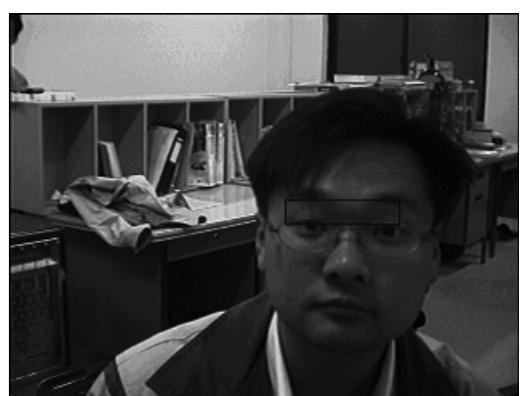
실제 영상 자료는 연구실을 배경으로 8mm 가정용 캠코더로 촬영된 영상을 대상으로 하였다. 캠코더로 촬영된 영상은 불규칙한 시간단위로 캡쳐되어 데이터로 사용되었다. 총 118개의 영상을 실험 대상으로 하였으며 이들 영상은 8~9개의 프레임으로 이루어진 14개의 집합으로 묶여졌다. 118개 중 23개의 영상이 안경을 쓰고 있는 상태였고 영상 중에는 빠른 속도로 전진하거나 이동하는 장면도 포함되었다. 수행 결과 118개의 실제 영상 중 95개가 눈을 추적하여 81%의 정확율을 보였다. 이들 23개의 잘못된 추적 중 20개는 successive fault 즉 이전 프레임에서 잘못 찾아진 영역을 기준으로 하여 연속적으로 발생한 오류였다. 또한, 잘못 추적된 결과 모두가 한쪽 눈만 찾는다거나 눈썹이나 눈과 눈썹 사이의 영역을 추적한 오류였다. 그림 4.14는 실제 영상에서 얻어진 잘못된 실험 결과의 예이다.



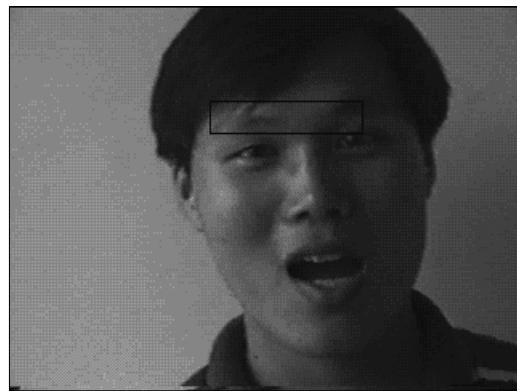
(a) 이전 프레임



(a) 현재 프레임



(b)



(c)



(d)

4.14 실험 결과 잘못 추출된 경우-실제 영상

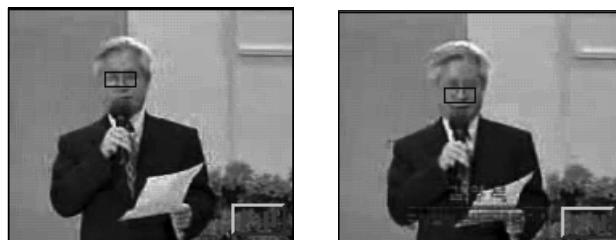
인터넷에서 취득된 영상은 총 151개이며 이중 72개의 영상이 안경을 쓴 것이었다. 이들 영상의 질은 매우 낮은 편이며 정면에서 옆면으로 얼굴을 잡자기 돌리거나 영상의 조명이 심하게 변하거나 배경이 복잡하고 눈이 아주 작아서 추출하기 힘든 경우가 모두 포함되어 있다. 이들 151개의 영상 중 119개의 영상이 정확히 눈을 추적하여 79%의 성공율을 나타내었다. 32개의 잘못된 영상 대부분이 눈썹이나 한쪽 눈 등을 찾았으며 일부는 코를 찾기도 하였다.

다음은 인터넷에서 취득된 영상 중에서 실험 결과가 잘못된 예이다.



(a) 이전 프레임

현재 프레임



(b) 이전 프레임

현재 프레임



(c) 이전 프레임

현재 프레임



(d) 이전 프레임

현재 프레임

4.15 실험 결과 잘못된 경우 - 인터넷 영상

영상 자료는 실제 영상 자료와 인터넷에서 취득된 영상 자료 및 쉽게 구할 수 있는 동영상 파일들에서 일정하지 않은 간격으로 캡쳐되었다.

실제 영상 자료들은 연구실을 배경으로 8mm 가정용 캠코더로 촬영된 영상을 대상으로 하였다. 총 118개의 영상을 실험 대상으로 하였으며 이들 영상은 8~9개의 프레임으로 이루어진 14개의 집합으로 둑여졌다. 118개 중

23개의 영상이 안경을 쓰고 있는 상태였고 영상 중에는 빠른 속도로 전진하거나 이동하는 장면도 포함되었다. 수행 결과 118개의 실제 영상 중 95개가 눈을 추적하여 81%의 정확율을 보였다. 이들 23개의 잘못된 추적 중 20개는 successive fault 즉 이전 프레임에서 잘못 찾아진 영역을 기준으로 하여 연속적으로 발생한 오류였다. 또한, 잘못 추적된 결과 모두가 한그림 4.14는 실제 영상에서 얻어진 잘못된 실험 결과의 예이다. 쪽 눈만 찾는다거나 눈썹이나 눈과 눈썹 사이의 영역을 추적한 오류였다.

인터넷에서 취득된 영상은 총 151개이며 이 중 72개의 영상이 안경을 쓴 것이었다. 이들 영상의 질은 매우 낮은 편이며 정면에서 옆면으로 얼굴을 갑자기 돌리거나 영상의 조명이 심하게 변하거나 배경이 복잡하고 눈이 아주 작아서 추출하기 힘든 경우가 모두 포함되어 있다. 이들 151개의 영상 중 119개의 영상이 정확히 눈을 추적하여 79%의 성공율을 나타내었다. 32개의 잘못된 영상 대부분이 눈썹이나 한쪽 눈 등을 찾았으며 일부는 코를 찾기도 하였다.

실험 결과 급격한 배경의 변화나 사람의 이동, 혹은 얼굴 형태의 변화 등에도 비교적 좋은 결과를 보였으며, 특히 눈이 너무 작거나 영상의 질이 낮은 경우에도 정확한 위치를 추적하였다. 오류의 수치가 높아진 원인은 첫 번째 프레임에서 잘못 찾아진 영역을 다시 템플릿화한 결과 이후 프레임에서 연속적으로 오류가 발생하였기 때문이었다. 이를 극복하기 위해 첫 번째 프레임에서 사용한 템플릿을 저장하고 있다가 이후에 찾아진 템플릿과 비교하는 등의 제어를 사용한다면 좀 더 좋은 결과를 보일 수 있을 것이다.

3.2 ART2를 이용한 눈위치 예측 모델

3.2.1 눈위치 예측 모델의 개요

Taken's embedding theorem에 의하면 사상 $F: R^{2m+1} \rightarrow R^{2m+1}$ 은 현재의 상태 $y(t)$ 에서 다음 상태 $y(t+1)$ 을 만들어낸다. 즉,

$$y(t+1) = F(y(t))$$
$$\begin{bmatrix} x(t+1) \\ \dots \\ x(t+1-2m) \end{bmatrix} = F \begin{bmatrix} x(t) \\ \dots \\ x(t-2m) \end{bmatrix}$$

여기서 m 은 예측에 쓰이는 차원이다.

예측하는 사상 $F^{\perp:R^{2m+1}} \rightarrow R$ 은 다음과 같이 나타낼 수 있다.

$$x(t+1) = F^\perp(x(t))$$
$$(여기서 x(t) = \begin{bmatrix} x(t-2m) \\ \dots \\ x(t-1) \\ x(t) \end{bmatrix} \circ \text{다.})$$

3.2.2 시공간 데이터의 공간화 spatio-temporal data의 spatial data화

눈의 자취는 시공간 정보를 포함하는 자료(spatio-temporal data)이다. 이를 그림으로 표현하면 다음과 같다. 즉, 시간 t 에서의 양쪽 눈의 중심을 frame t (x, y)라고 하면 연속되는 여덟 개의 프레임에서 눈의 중심을 frame1(x, y), frame2(x, y), frame3(x, y), frame4(x, y), frame5(x, y), frame6(x, y), frame7(x, y), frame8(x, y)로 표현할 수 있다. 따라서, 눈의 궤적은 각 프레임에서의 눈의 중심을 순차적으로 연결한 frame1(x, y) \rightarrow frame2(x, y) \rightarrow frame3(x, y) \rightarrow frame4(x, y) \rightarrow frame5(x, y) \rightarrow frame6(x, y) \rightarrow frame7(x, y) \rightarrow frame8(x, y)가 된다.



frame1(x,y)

frame2(x,y)

frame3(x,y)

frame4(x,y)



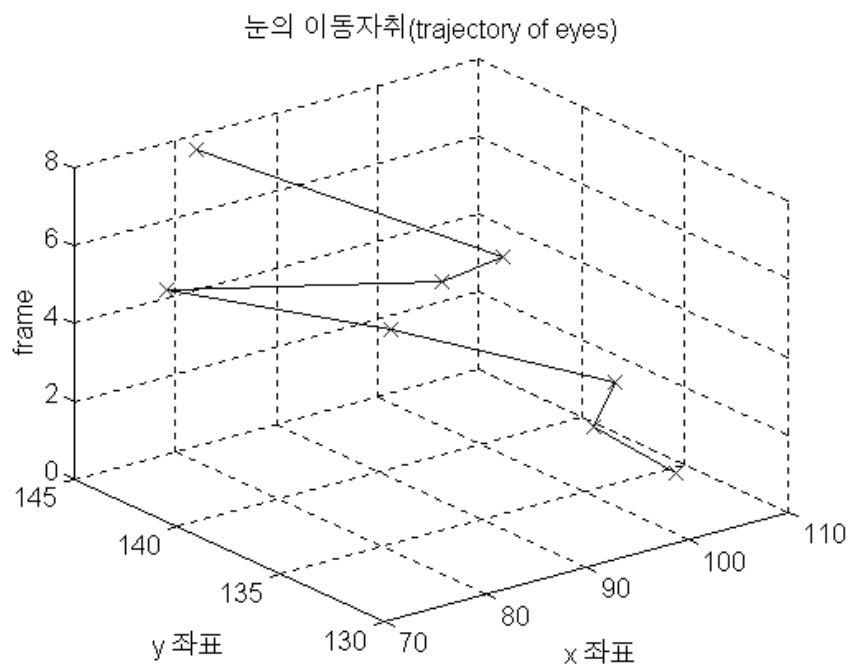
frame5(x,y)

frame6(x,y)

frame7(x,y)

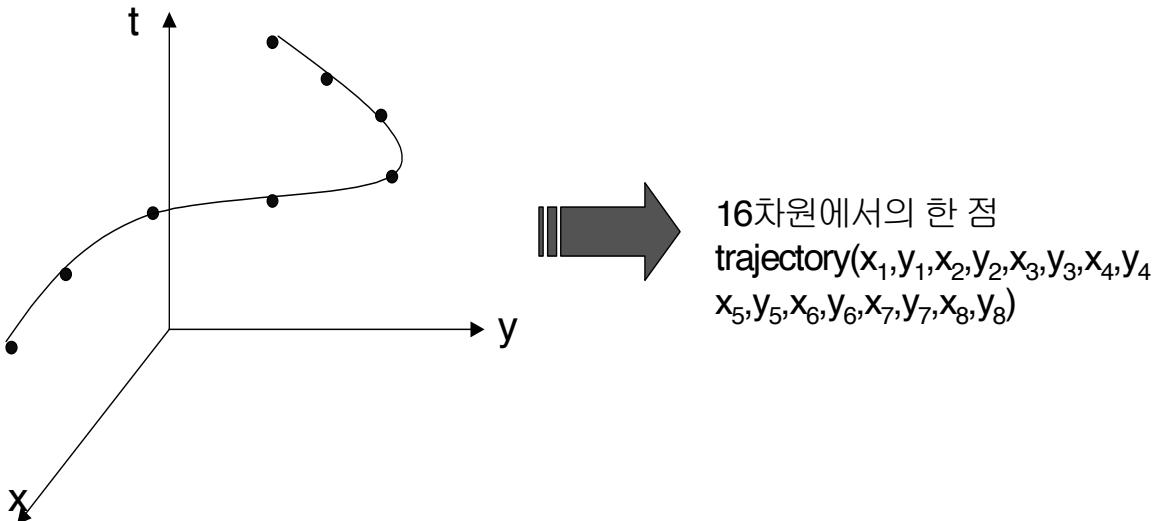
frame8(x,y)

그림???에서의 눈의 자취를 보기 쉽게 그림???에 그래프로 나타내었다. 여기서 x좌표, y좌표는 해당 frame의 눈중심의 x,y좌표를 의미한다.



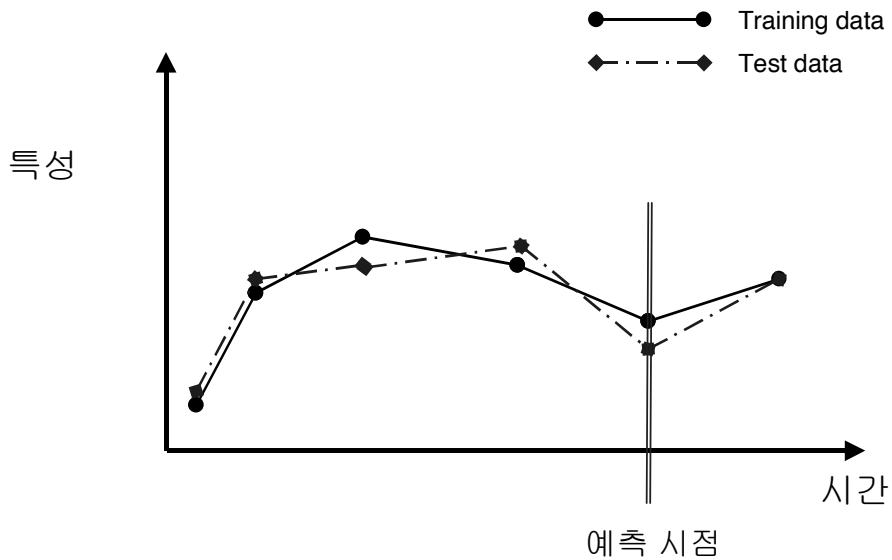
본 연구에서는 이와같은 시공간 데이터를 공간데이터로 변환하여 사용하였다. 아

래의 그림????과 같이 (x, y) 속성을 가진 여덟 개의 프레임을 연결하는 궤적은 16차원상의 한 점 trajectory($x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4, x_5, y_5, x_6, y_6, x_7, y_7, x_8, y_8$)로 나타낼 수 있다. 이 점은 16차원상에서 단 하나만 존재하기 때문에 다른 경로를 가진 궤적들과 완전히 구별된다. 이렇게 공간화된 궤적은 ART2 신경망의 입력벡터로 사용된다.

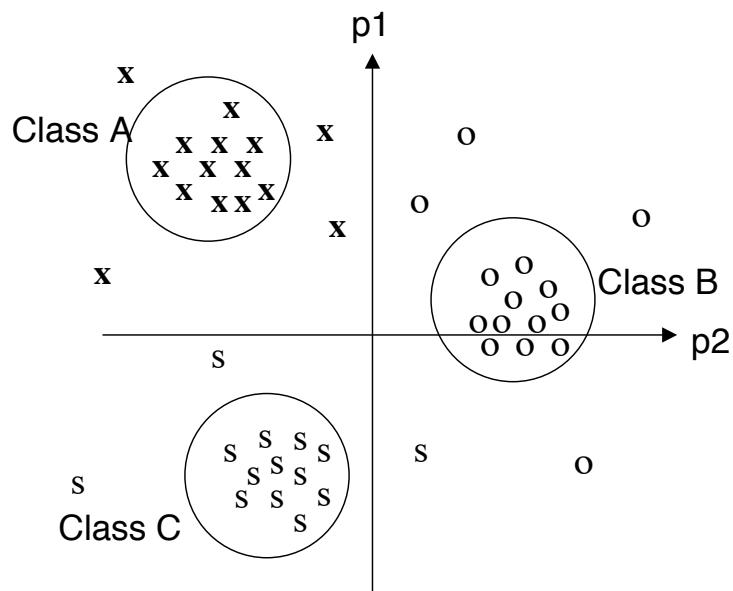


3.2.3 ART2 Neural Network

본 절에서는 앞절에 설명된 입력벡터를 가지고 다음 움직임을 예측하는 ART2 신경망에 대해 기술한다. 본 연구에서는 움직임 예측을 위해 ‘유사한 경로를 가진 두 궤적은 다음 위치(속도) 역시 유사하다.’라는 조건을 만족한다고 가정한다. 그림 ???는 특성이 하나이고 6개의 프레임으로 구성된 데이터를 그래프로 나타낸 것이다. 다음 프레임에서의 예측위치를 알아야 하는 실험 데이터와 다음 프레임에서의 위치를 이미 알고 있는 훈련 데이터가 예측시점 까지의 5개 프레임이 비교적 유사한 경로를 가졌을 때, 다음 프레임에서의 위치도 유사할 것으로 예측한다.

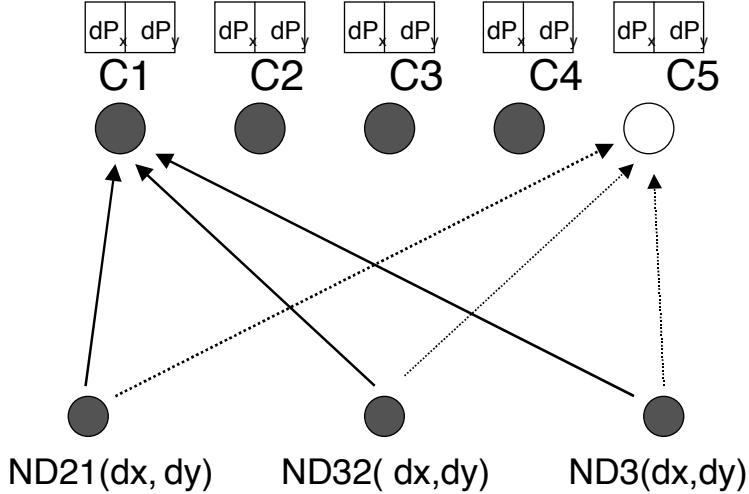


앞절에서 언급한 시공간 데이터의 공간화에 의해 움직임의 궤적을 다차원상의 유일한 한 점으로 나타낼 수 있다고 기술하였다. 이렇게 공간상에서 한 점으로 나타내어진 궤적은 그림???? 와 같이 분류될 수 있다. 분류 기준은 원의 중심인 seed점으로부터 거리이며 기준 거리에 따라 예측의 정확도가 달라진다. 이것은 실험 및 결과 부분에서 자세하게 다룬다.



ART 신경망은 무감독 신경망(unsupervised neural network)의 한 종류로서 입력의 패턴에 따라 클래스를 분류한다. 따라서, 다차원공간상에서 하나의 점으로 나타내어

진 케적을 분류하는 신경망으로 매우 적합하다. 본 연구에서는 ART2 아키텍처를 사용하여 케적을 클래스별로 분류하였다. 그림???는 본 연구에서 사용한 ART2의 구조이다.



P_x, P_y : predicted x , y value of eye center in next frame

$C_i = i^{\text{th}}$ class

$ND_{ij}(dx,dy)$ { $dx = (x \text{ of } i^{\text{th}} \text{ frame} - x \text{ of } j^{\text{th}} \text{ frame})/\text{width}$
 $dy = (y \text{ of } i^{\text{th}} \text{ frame} - y \text{ of } j^{\text{th}} \text{ frame})/\text{height}$
: Normalized Difference

가. 구현된 ART2의 구조

구현된 신경망은 2층의 완전히 연결된 구조를 갖는다. 입력뉴런의 개수는 $N_{\text{properly}} \times (N_{\text{frame}} - 1)$ 이다. 여기서 N_{properly} 는 눈의 중심을 나타내는 속성들의 개수이고 N_{frame} 은 클러스터링에 사용되는 프레임의 개수이다. 움직임의 상대위치인 속도로 입력벡터를 구성하기 때문에 원래 N_{frame} 이었던 프레임의 수가 $(N_{\text{frame}} - 1)$ 으로 줄어든다. 또한, 영상의 크기에 종속받지 않기 위해서 영상의 넓이와 높이로 정규화시키기 위해 *width*, *height*라는 변수를 두어 영상의 넓이와 높이값을 저장한다. 출력 뉴런은 클래스를 의미한다. 입력 뉴런에서 출력뉴런으로 연결된 가중치는 각 클래스의 seed값이다. 입력벡터가 들어가면 가장 거리가 짧고 유사한 클래스에 해당되는 출력 뉴런 하나만 활성화시키고 활성화된 출력 뉴런에 연결된 가중치를 훈련시킨다. 가장 유사한 출력 뉴런에 대한 확인과정에서 vigilance 값을 초과하게 되면 새로운 출력 뉴런이 생성되므로 출력 뉴런의 수는 가변적이다. 각 클래스(출력 뉴런)는 현재 프레임과 다음 프레임간의 예측속

도를 저장하는 저장공간 P_x, P_y 을 가진다.

나. 알고리즘

1) 입력 전처리와 초기화

앞장에서 구현된 동영상에서의 눈동자 추출 알고리즘으로 얻어진 눈동자 궤적에 관한 정보는 다음과 같다. k번째 궤적을 $T_k\{W, H, x_1, x_2, \dots, x_{N-1}, x_N, y_1, y_2, \dots, y_{N-1}, y_N\}$ 라 한다. 여기서 N 은 k번째 궤적을 구성하는 영상 프레임의 개수, W 는 k번째 궤적의 영상 넓이, H 는 k번째 궤적의 영상 높이이고 x_j 는 k번째 입력 집합들 중 j 번째 프레임에서의 눈동자 중심 위치의 x 좌표, y_j 는 j 번째 프레임에서의 눈동자 중심 위치의 y 좌표이다. 이렇게 눈추출 모듈에서 얻어진 눈동자의 궤적을 신경망의 입력으로 사용하기 위해 다음과 같이 변환한다.

$$M = N - 1$$

$$ND_k\{ ND_1(x), ND_2(x), \dots, ND_{M-1}(x), ND_M(x), ND_1(y), ND_2(y), \dots, ND_{M-1}(y), ND_M(y) \}$$

$$ND_j(x) = \frac{x_{j+1} - x_j}{W}, \quad ND_j(y) = \frac{y_{j+1} - y_j}{H} \quad (1 \leq j \leq M) \quad (\text{식}???)$$

여기서 맨 마지막 프레임 $ND_M(x), ND_M(y)$ 는 예측되어야 할 프레임 $ND_{pred}(x), ND_{pred}(y)$ 으로 사용한다. 따라서 위의 식??은 다음 식??과 같이 바뀔 수 있다.

$$M = N - 2$$

$$ND_k\{ ND_1(x), ND_2(x), \dots, ND_M(x), ND_{pred}(x), ND_1(y), ND_2(y), \dots, ND_M(y), ND_{pred}(y) \}$$

식(???)에 의해 영상은 프레임간의 속도 벡터를 넓이와 높이로 정규화시킴으로써 영상의 크기와 눈동자의 절대위치에 무관하도록 설정하였다.

클러스터는 임의로 정의된 수만큼 구조체 형태로 설정되어 있고 입력 뉴런에서 출력 뉴런으로 가는 가중치들은 실수 타입의 배열형태로 존재한다. 클러스터 정보는 클러스터의 사용여부를 나타내는 플래그와 중심 벡터와 예측 벡터가 있다. 이들 정보는 모두 0으로 초기화된다. 최초의 클러스터는 첫번째 입력 벡터값으로 클러스터의 중심좌표와 예측 벡터값을 설정하게 되며 계속적으로 들어오는 입력 벡터값과 기존의 클러스터간의 거리에 따라 이전 클러스터에 벡터를 할당하거나 새로운 클러스터를 생성한다. 새로운 클러스터가 생성되는 경우에는 클러스터를 생성하게 한 입력 벡터의 좌표와 멤버값을

새로운 클러스터에 해당하는 가중치로 설정한다.

```
struct CLUSTERSTR
{
    int use;
    float count;
    struct _point predict;
}
```

2) 훈련 과정

ART 네트워크는 기본적으로 입력 벡터와 가장 유사한 출력 벡터 하나만을 골라 해당 가중치를 변경하는 winner-take-all 전략을 취한다. 이로 인해 신경망은 전체 출력 뉴런 가중치를 조정해야 하는 부담에서 벗어난다. 각 클러스터에 대한 가중치는 클러스터의 중심 좌표를 의미하며 클러스터의 정보에는 예측 벡터와 클러스터에 해당되는 데이터의 수가 있다.

클러스터의 중심좌표와 현재 입력된 눈동자의 궤적간의 거리는 다음과 같은 수식???을 이용하여 구한다.

$$D_i = \sum_{j=1}^{j=M} (\rho + T \times j) \sqrt{(W_{ij}(x) - ND_j(x))^2 + (W_{ij}(y) - ND_j(y))^2} : \text{distance (식???)}$$

D_i : 출력 뉴런 i 의 중심좌표와 입력 뉴런입력된 궤적간의 거리

$W_{ij}(x)$: 클러스터 i 와 현재 입력된 j 번째 프레임의 x 좌표간의 가중치

$W_{ij}(y)$: 클러스터 i 와 현재 입력된 j 번째 프레임간 y 좌표간의 가중치

ρ : 근접 시간에 대한 초기 가중치 ($0 < \rho \leq 1$)

T : 근접 시간에 대한 증가치 ($0 < T < 1$)

winner 뉴런은 각 출력 뉴런에 연결된 가중치와 현재 입력 뉴런간의 거리중 가장 짧은 출력 뉴런이 선택된다. 식??? 은 winner 뉴런의 선택을 수식으로 보여준다.

$$WC = \min(D_i) : \text{winner cluster (식???)}$$

winner로 선택된 출력 뉴런은 vigilance test를 거치게 된다. vigilance는 특정 입력값과 출력뉴런에 연결된 가중치간의 유사도를 의미하며 이 값이 임의로 임계치 θ 보다 크면 입력값과 가중치간의 차이가 지나치게 큰 것이므로 출력 뉴런을 비활성화시키고 적절한 다른 뉴런을 찾는다. vigilance 조건을 만족하는 뉴런이 없으면 새로운 뉴런을 생성한다.

$$V_i = \sum_{j=1}^{j=M} |W_{ij} - ND_j| : \text{vigilance}$$

식 (**)는 가중치 x 에 대한 훈련 과정을 수학식으로 표기한 것이다. 마찬가지로 가중치 y 와 μ 에 대하여 훈련을 반복한다.

3) 목표 속도 벡터 추출

$$WC_{pred}(x)^k = ND_{pred}^1 + \sum_{i=2}^{i=k-1} \frac{ND_{pred}(x)^i - WC_{pred}(x)^i}{\sigma \times WC_{count}^i} \quad (3 \leq i \leq dataSize) \quad (\sigma > 1)$$

$$WC_{pred}(y)^k = ND_{pred}^1 + \sum_{i=2}^{i=k-1} \frac{ND_{pred}(y)^i - WC_{pred}(y)^i}{\sigma \times WC_{count}^i} \quad (3 \leq i \leq dataSize) \quad (\sigma > 1)$$

$$WC_j^k = WC_j^1 + \sum_{i=2}^{i=k-1} \frac{ND_j^i - WC_j^i}{WC_{count}^i \times \sigma} \quad (3 \leq i \leq dataSize), (1 \leq j \leq M), (\sigma > 1)$$

4) Error measure

$$NMSE = \frac{\sum_{j=1}^{j=K} \sqrt{(ND_{pred}(x)^j - WC_{pred}(x)^j)^2 + (ND_{pred}(y)^j - WC_{pred}(y)^j)^2}}{K}$$

K : 현재까지 입력된 프레임 집합의 개수

4. 실험 및 결과

영상자료는 실제 영상 자료와 인터넷에서 취득된 영상 자료 및 쉽게 구할 수 있는 동영상 파일들에서 일정하지 않은 간격으로 캡쳐되었다.

실제 영상 자료들은 연구실을 배경으로 8mm 가정용 캠코더로 촬영된 영상을 대상으로 하였다. 총 118개의 영상을 실험 대상으로 하였으며 이들 영상은 8~9개의 프레임으로 이루어진 14개의 집합으로 묶여졌다. 118개 중 23개의 영상이 안경을 쓰고 있는 상태였고 영상 중에는 빠른 속도로 전진하거나 이동하는 장면도 포함되었다. 수행 결과 118개의 실제 영상 중 95개가 눈을 추적하여 81%의 정확율을 보였다. 이들 23개의 잘못된 추적 중 20개는 successive fault 즉 이전 프레임에서 잘못 찾아진 영역을 기준으로 하여 연속적으로 발생한 오류였다. 또한, 잘못 추적된 결과 모두가 한쪽 눈만 찾는다거나 눈썹이나 눈과 눈썹 사이의 영역을 추적한 오류였다. 그림 4.14는 실제 영상에서 얻어진 잘못된 실험 결과의 예이다.

인터넷에서 취득된 영상은 총 151개이며 이 중 72개의 영상이 안경을 쓴 것이었다. 이들 영상의 질은 매우 낮은 편이며 정면에서 옆면으로 얼굴을 갑자기 돌리거나 영상의 조명이 심하게 변하거나 배경이 복잡하고 눈이 아주 작아서 추출하기 힘든 경우가 모두 포함되어 있다. 이들 151개의 영상 중 119개의 영상이 정확히 눈을 추적하여 79%의 성공율을 나타내었다. 32개의 잘못된 영상 대부분이 눈썹이나 한쪽 눈 등을 찾았으며 일부는 코를 찾기도 하였다.

5. 결론