[UCI]I804:24011-200000238394

Collection @ chosun

2009年　8月

博士學位論文

수화인식을 위한 손 형상 인식 및 가상환경 제어

朝鮮大學校 大學院

電算統計學科

金　昇　彦

Collection @ chosun

수화인식을 위한 손 형상 인식 및 가상환경 제어

Hand Shape Recognition for Sign Language and Virtual Reality Control

2009년 8월 25일

朝鮮大學校 大學院

電算統計學科

金 昇 彦

Collection @ chosun

수화인식을 위한 손 형상 인식 및 가상환경 제어

指導敎授　丁 采 榮

이 論文을 理學博士學位 申請 論文으로 提出함.

2009년 4월 17일

朝鮮大學校 大學院

電算統計學科

金 昇 彦

Collection @ chosun

金昇彦의 博士學位論文을 認准함

委 員 長　朝鮮大學校 敎授　이 윤 기 印

委　　員　草堂大學校 敎授　정 남 채 印

委　　員　朝鮮大學校 敎授　배 상 현 印

委　　員　朝鮮大學校 敎授　권 용 만 印

委　　員　朝鮮大學校 敎授　김 채 영 印

2009年　6月

朝鮮大學校　大學院

Collection @ chosun

목 차

표 목 차 ····· iv
그림 목차 ····· v
ABSTRACT ····· x

Ⅰ. 서 론 ····· 1
1. 연구배경 ····· 1
2. 기존의 주요 손 인식 방법 ····· 4
2-1. 손 영역 검출 관련 연구 ····· 5
2-2. 수화 인식을 위한 손 형상 인식 관련 연구 ····· 14
3. 연구의 목적 및 논문의 구성 ····· 27

Ⅱ. 신경망 알고리즘 ····· 30
1. 신경망의 개요 ····· 30
2. 신경망의 생물학적 구조 ····· 32
3. 뉴런의 모델링 ····· 35
4. 신경망 모델의 구조 및 장·단점 ····· 36
5. 신경망 알고리즘에 대한 관련 연구 ····· 39
5-1. Hebb의 학습 ····· 39
5-2. 퍼지 최대-최소 신경망 알고리즘 ····· 41
5-3. 퍼셉트론(Perceptron) ····· 44
5-4. 다층 퍼셉트론(Multilayer Perceptron) ····· 46
5-5. 역전파 알고리즘(Back Propagation Algorithm) ····· 48
5-6. RBF 신경망 ····· 49

Collection @ chosun

5-7. LVQ 신경망 ····· 51

Ⅲ. 피부색 정보를 이용한 손 영역 검출 ····· 54
1. Skin-Tone에 의한 세그멘테이션 ····· 54
2. 손 영역 추출 ····· 57
2-1. 레이블링(Labeling) ····· 58
2-2. Mathematical Morphology ····· 61

Ⅳ. 신경망을 이용한 손 형상 인식 ····· 68
1. 제안된 알고리즘 순서도 ····· 68
2. 손 영상에 대한 전처리 ····· 69
2-1. 손 영상의 크기 및 색상정보 정규화 ····· 70
3. 템플리트 매칭(Template Matching) ····· 72
3-1. 템플리트 매칭의 개요 ····· 72
3-2. 템플리트 매칭을 이용한 가상환경 제어 ····· 81
4. 신경망을 이용한 손 형상 인식 ····· 86
4-1. 제안한 신경망 알고리즘 ····· 86
4-2. SOFM의 학습 과정과 장·단점 ····· 89
4-3. SOFM의 학습 단계 ····· 91

Ⅴ. 실험결과 ····· 96
1. 템플리트 매칭에 의한 가상환경 제어 ····· 96
1-1. 템플리트에 의한 인식률 및 실험 결과 ····· 97
1-2. 가상환경에서의 응용 및 실험 결과 ····· 100
2. 신경망에 의한 손 형상 인식 ····· 101
2-1. 신경망에 의한 손 형상 인식 결과 ····· 101

Collection @ chosun

2-2. 손 형상 인식의 응용 및 실험 결과 ····· 106

VI. 결론 및 발전 계획 ····· 109

참 고 문 헌 ····· 112

Collection @ chosun

표 목 차

[표 1-1] Horn의 Optical Flow 계산 알고리즘 ····· 8
[표 2-1] 뉴런의 구조 ····· 34
[표 2-2] 생물학적 신경망의 특성 ····· 34
[표 2-3] 신경 시스템과 신경회로망 시스템의 비교 ····· 35
[표 2-4] 신경망과 디지털 컴퓨터의 비교 ····· 38
[표 2-5] 신경망의 응용 분야 ····· 38
[표 2-6] FMMNN의 학습과정 ····· 42
[표 2-7] 다층 퍼셉트론의 학습 절차 ····· 47
[표 2-8] 역전파 알고리즘의 단계별 처리 ····· 49
[표 2-9] LVQ 알고리즘 ····· 53
[표 3-1] Pseudo 코드 ····· 60
[표 4-1] 손의 회전 방향 표시 ····· 83
[표 4-2] SOFM의 학습 3단계 ····· 91
[표 4-3] SOFM 알고리즘 ····· 94
[표 5-1] 배경에 따른 손가락 끝 점 추출 인식률 ····· 98
[표 5-2] 배경에 다른 손 영역 추출 ····· 99
[표 5-3] [그림 5-1]의 방향정보를 이용한 방향전환 인식 결과 ····· 101
[표 5-4] 단순한 배경과 복잡한 배경에서의 손 형상 인식률 ····· 104
[표 5-5] 전체 모델 영상에 속하는 영상과 속하지 않는 영상의 인식률 ····· 104
[표 5-6] 기존 알고리즘과 비교한 성능평가 분석표 ····· 105

Collection @ chosun

그림 목차

[그림 1-1] 손 제스처의 해석과 인식 ····· 5
[그림 1-2] 데이터 글로브 종류 ····· 6
[그림 1-3] 특징 벡터를 형상 구분의 척도로 사용하는 방법 ····· 6
[그림 1-4] Color 정보에 의한 손 영역 추출 ····· 8
[그림 1-5] x, y, z 방향에 따른 손 동작의 예 ····· 9
[그림 1-6] 각 방향의 Optical Flow ····· 9
[그림 1-7] 각 방향의 손 영역 추출 결과 ····· 9
[그림 1-8] 차 영상을 이용한 영상 분할 ····· 10
[그림 1-9] 각 Gesture에 대한 모델링 ····· 10
[그림 1-10] 얼굴과 손 영역 추출 ····· 11
[그림 1-11] 추출한 영역에 대한 히스토그램 ····· 11
[그림 1-12] Static 영역과 Motion 영역 ····· 11
[그림 1-13] 손의 움직임에 따른 Tracking 결과 ····· 12
[그림 1-14] 특징벡터(Feature Vector)와 히스토그램(Histogram) ····· 12
[그림 1-15] 방향에 따른 히스토그램 ····· 13
[그림 1-16] 명령 어휘 Gesture ····· 14
[그림 1-17] Matrix Display ····· 14
[그림 1-18] 6가지의 손 모양 ····· 15
[그림 1-19] 물체 인식을 위한 무게중심 프로필 ····· 15
[그림 1-20] 무게중심 프로필이 다중값을 갖는 경우 ····· 16
[그림 1-21] [그림 1-18]의 무게중심 프로필 ····· 17
[그림 1-22] 무게중심을 이용한 수화인식 시스템 ····· 17
[그림 1-23] 6가지 수화동작에 대한 영역 추출 및 무게중심 프로필 ····· 17
[그림 1-24] 인식에 사용된 수화 단어(9가지) ····· 18

Collection @ chosun

[그림 1-25] 3계층적 9진 트리의 구성 ····· 19
[그림 1-26] 체인코드와 HMM 모델 ····· 19
[그림 1-27] HMM에 의한 인식 ····· 19
[그림 1-28] 마커 위치 ····· 20
[그림 1-29] 마커의 중심 찾기 ····· 21
[그림 1-30] 마커 수의 변화 ····· 21
[그림 1-31] 손의 관절구조 및 마커번호 ····· 22
[그림 1-32] 마커의 움직임 추적 ····· 22
[그림 1-33] 마커의 이동 추적 ····· 23
[그림 1-34] 동일마커 추적 방법 ····· 23
[그림 1-35] 미국 수화의 표준으로 사용되는 알파벳 손가락문자(a~y) 영상 ····· 24
[그림 1-36] 참조 영상 ····· 25
[그림 1-37] 상위도 공간에서 4장의 영상을 배치시킨 모습 ····· 25
[그림 1-38] 상위도에 따른 영상들의 분포 ····· 26
[그림 1-39] 카메라로 획득한 영상과 가속도 센서 보드 ····· 26
[그림 1-40] 손 영역 추출 ····· 27
[그림 1-41] 왼손과 오른손 추출 ····· 27
[그림 2-1] PE(Processing Element)의 기본 구조와 신경망 ····· 31
[그림 2-2] 예제를 통한 신경망 구성 ····· 32
[그림 2-3] 생물학적 신경세포의 구조 ····· 33
[그림 2-4] 생물학적 신경망의 추상화 ····· 36
[그림 2-5] 뉴런 모델 ····· 36
[그림 2-6] Hebbian Learning의 개념 ····· 39
[그림 2-7] Hebb의 학습모델 ····· 40
[그림 2-8] Hebb의 학습 순서도 ····· 40
[그림 2-9] 교사학습과 무교사학습 ····· 41

Collection @ chosun

[그림 2-10] 초상자들의 경계 형성 ····· 42
[그림 2-11] 두 초상자의 거리가 먼 경우 테스트 데이터에 대한 각 초상자의 멤버쉽 값 비교 ····· 43
[그림 2-12] 두 초상자의 거리가 가까운 경우 테스트 데이터에 대한 각 초상자의 멤버쉽 값 비교 ····· 43
[그림 2-13] 퍼셉트론 모델 ····· 44
[그림 2-14] 퍼셉트론의 학습과정 ····· 45
[그림 2-15] 퍼셉트론의 한계 ····· 45
[그림 2-16] 다층 퍼셉트론의 모델 ····· 46
[그림 2-17] 다층 퍼셉트론의 학습과정 ····· 46
[그림 2-18] 다층 퍼셉트론의 순서도 ····· 47
[그림 2-19] BPN의 계산과 학습 방향 ····· 49
[그림 2-20] RBF 신경망의 구조 ····· 51
[그림 2-21] LVQ 신경망의 구조 ····· 52
[그림 3-1] RGB Color Model ····· 56
[그림 3-2] YUV Color Model ····· 56
[그림 3-3] 입력 영상의 손 영역 추출 ····· 57
[그림 3-4] 이웃화소의 예 ····· 57
[그림 3-5] 레이블링(Labeling)된 2진화상 ····· 58
[그림 3-6] 순차 연결성분 알고리즘 ····· 61
[그림 3-7] 2진 영상에서의 점집합의 예 ····· 62
[그림 3-8] 전형적인 구조적 요소의 몇 가지 예 ····· 62
[그림 3-9] 1차원 신호의 집합 및 Reflection set ····· 63
[그림 3-10] Dilation과 Erosion 연산의 예 ····· 64
[그림 3-11] 실제 영상에 적용한 FSP Dilation과 FSP Erosion 연산의 예 ····· 64
[그림 3-12] 손 영역 노이즈 제거 및 보정 ····· 67

$\overline{B}$

Collection @ chosun

[그림 4-1] 제안된 알고리즘 순서도 ····· 68
[그림 4-2] 템플리트 매칭과 손 형상 인식을 위한 전처리 과정 ····· 69
[그림 4-3] 손가락 끝 점을 인식하기 위한 모델영상의 크기 및 2진화 ····· 70
[그림 4-4] 손 형상 인식을 위한 영상의 크기 및 Gray Level 변환 ····· 70
[그림 4-5] 손 형상 인식을 위한 모델 영상(모델 영상의 일부) ····· 71
[그림 4-6] 탐색할 영상과 템플리트 모델 영상 ····· 72
[그림 4-7] 템플리트 매칭 비교 수행과정 ····· 73
[그림 4-8] 밝은 곳에서 촬영된 검사영상 ····· 75
[그림 4-9] 두 영상의 비교 ····· 76
[그림 4-10] 영상의 밝기 정규화 ····· 77
[그림 4-11] 다차원 공간에서 영상 데이터를 표현하는 벡터의 정의 ····· 78
[그림 4-12] 샘플링과 양자화 ····· 79
[그림 4-13] 평면상의 두 벡터 ····· 79
[그림 4-14] 방향 제어를 위한 시스템 구성 ····· 82
[그림 4-15] 손의 회전 방향 표시 ····· 83
[그림 4-16] 템플리트 매칭 수행과정 ····· 84
[그림 4-17] 템플리트 매칭을 이용한 손가락 끝 점 인식 ····· 84
[그림 4-18] 손의 방향 인식 ····· 86
[그림 4-19] 손가락 끝 점의 위치변환에 따른 각의 변화 ····· 86
[그림 4-20] SOFM 신경망의 구조 ····· 87
[그림 4-21] 전체 모델 영상의 평균(모델 영상은 일부만 표시) ····· 88
[그림 4-22] 입력 특징 벡터와 모델 영상의 특징 벡터간의 경쟁과정 ····· 89
[그림 4-23] 협동과정 ····· 90
[그림 4-24] 승자 뉴런에 인접한 이웃 뉴런 ····· 90
[그림 4-25] 적응과정에 의한 연결 강도의 갱신 ····· 91
[그림 4-26] 입력영상과 모델영상의 거리 ····· 92

Collection @ chosun

[그림 4-27] SOFM 순서도 ····· 95
[그림 5-1] 실험에 사용된 손 형상 ····· 97
[그림 5-2] 방향전환에 따른 각도의 변화 그래프 ····· 97
[그림 5-3] 실험을 위한 10개의 손 형상 ····· 98
[그림 5-5] 복잡한 배경에서의 손 영역 추출 및 한 손가락 인식 ····· 99
[그림 5-4] 복잡한 배경에서의 손 영역 추출 및 다수의 손가락 인식 ····· 99
[그림 5-6] 가상공간에서의 방향제어 실험 결과 ····· 100
[그림 5-7] 최상위 모델로 인식된 손 형상과 인식된 손 형상의 수화 의미 ····· 102
[그림 5-8] 손 형상에 대한 수화 의미 ····· 103
[그림 5-9] 모델 영상에 속하지 않는 손 형상 ····· 104
[그림 5-10] 모델 영상에 속하지 않는 영상에 대한 인식 결과 ····· 105
[그림 5-11] 수화인식을 위한 손 형상 인식 시스템 ····· 107
[그림 5-12] 복잡한 배경에서 인식 오류 ····· 107
[그림 5-13] 복잡한 배경에서의 인식 ····· 108
[그림 5-14] 모델에 속하지 않는 영상의 오-인식 ····· 108

Collection @ chosun

ABSTRACT

Hand Shape Recognition for Sign Language and Virtual Reality Control

Seung-Eon Kim
Advisor : Prof. Chae-Young Jung
Department of Computer Science and Statistics,
Graduate School of Chosun University

As the necessity of smooth interaction between human and computer grows up among the environment of user-friendly interface implementation, there are many researches analyzing human actions. In order to transmit information, human beings use, besides language, nonverbal communication methods such as motions, facial expressions and others. If we can analyze these nonverbal methods accurately, we can build a natural and intellectual interface between human and computer. Human beings use various methods in order to transmit their own intention to the others. In this case we easily think of verbal methods, but many researches show that nonverbal communication methods such as gestures, motions of the hand, facial expressions and others transmit more information than language. In order to manufacture more natural and intellectual system, we need to automate the nonverbal communication methods. Above everything else, the hand has been one of the most important

Collection @ chosun

instruments of human beings and, on the point that recognizing the meanings of its motion and shape leads to the embodiment of the natural interface between the information system and human beings, it has long been treated as research themes. But it is difficult to extract the shape of human hands accurately among complex environment and difficult to use it as meaningful input. In order for this, algorithms for accurate hand shape segmentation and the fast processing speed for the analysis and recognition of hand shape are basically necessary.

In this paper, using single camera without any sensor and taking the hand shape as input information, we try to control the virtual environment through the recognition of fingertip point by template matching. And using nerve network algorithms, we aim to embody a hand shape recognition system that is stable and strong for sign language recognition. This system will be able to provide a more convenient and natural interface between the disabled and the general people. In order to communicate with the disabled, the general people should learn the sign language. But it is inefficient in economical points and in matters of time. This paper tries to provide the interface for the disabled and the general people in order to let them communicate conveniently without those inefficiencies. Our proposed system, taking the image through the camera as input and extracting the hand shape from the scene, recognizes the extracted hand shape. (Before this process, we formalized the information about the size, compression and color of mode image.) By controlling the virtual environment through the recognition of fingertip point by template matching and by applying the nerve network algorithms on the recognized hand shape information, we have improved the stability and the accuracy about the recognition ratio of hand shape.

Collection @ chosun

I. 서론

1. 연구배경

최근 사용자 친화형 유저 인터페이스 구현 환경이 대두되면서 인간과 컴퓨터간의 원활한 상호작용의 필요성이 커짐에 따라 영상처리를 이용하여 인간의 행동을 분석하는 연구가 많이 수행되고 있다[1]. 인간은 정보전달을 위하여 언어 이외에 동작, 표정과 같은 비언어적인 수단을 이용한다. 이러한 비언어적인 수단을 정확히 분석 할 수 있다면 인간과 컴퓨터간의 자연스럽고 지적인 인터페이스를 구축할 수 있을 것이다.

인간은 다양한 수단을 사용하여 상대방에게 자신의 의사를 전달한다. 이 경우 흔히 언어적 수단만을 고려하기 쉬우나 이 외에도 몸짓, 손짓, 얼굴의 표정 등과 같은 비언어적 수단이 보다 많은 정보를 전달한다는 연구 결과가 보고되어 있다[2]. 인간과 의사소통이 자유스럽고 보다 지적인 시스템을 제작하기 위해서는 무엇보다도 이러한 비언어적 대화 수단을 자동화함이 필요하다. 그 중에서도 손은 인간에게 있어서 가장 중요한 도구의 하나로 그 동작과 모양이 의미하는 바를 인식하는 것은 정보 시스템과 인간 사이의 자연스러운 인터페이스를 실현할 수 있다는 점에서 오랜 기간 동안 많은 사람들의 연구테마로 다루어져 왔다. 그러나 복잡한 환경으로부터 사람의 손 형상만을 정확히 추출하고 인식하여 의미 있는 입력정보로 사용하는 것은 어려운 일이다. 기본적으로 정확한 손 영역 분할을 위한 알고리즘이 필수이며 손 형상의 분석과 인식을 위한 빠른 처리 속도 또한 고려해야할 요소이다. 기존의 연구들은 크게 손 형상을 분석하기 위해 Data Glove를 사용하는 방법[2]과 3차원 모델을 사용하거나 스테레오 비전을 이용하여 추출된 손의 3차원 기하학적인 정보를 분석하는 방법[3], 손의 2차원 패턴 정보를 분석하는 방법[4]으로 분류할 수 있다. Data glove를 사용하는 방법은 정밀도가 낮고 전체 시스템과 접속하기 위해 반드시 연결선이 필요하다는 점에서 자연스러운 인터페이스 구축에는 많은 문제점을 안고 있다. 3차원 기하 정보를 이용하는 방법은 스테레오 영상을 이용한 3차원 정보추출과 복잡한 손의 3차원 운동 모델을 구축하기 위한 많은 계산량이 필요하므로 고성능의 하드웨어가 요구된다는 단점이 있다. 2차원 패턴 정보를 이용하는

Collection @ chosun

방법들은 사람의 피부색 컬러분포를 사전 학습과정을 통해 계산하여 입력영상 내에서 피부색 컬러분포에 해당되는 영역만을 추출하는 방법을 적용하였으나 대부분 단일 배경이거나 조명 변화가 거의 없다고 가정하였다.

손에 별도의 센서를 장착하지 않고 손 형상을 인식하기 위해서는 시각적인 방법을 이용하는 수 밖에 없다. 그러나 시각적인 방법은 모델구축의 어려움과 세그멘테이션의 어려움 때문에 알기 쉬운 표식을 붙인다거나[5] 손의 형상이 뚜렷이 분리될 수 있는 간단한 문제를 위주로 적용되어 왔다. 또 난이도와 해상도의 문제 때문에 손가락 형상의 자세한 인식보다는 상반신을 포함한 어깨 전체의 제스처인식을 연구 목표로 삼고 있는 경우가 많다[6].

손 형상 인식의 대표적인 방법으로 화소간의 휘도와 경도를 구해 히스토그램을 만들고 이 히스토그램을 평활화하여 얻어진 특징 벡터를 형상 구분의 척도로 삼는 방법이 있다[7]. 즉 모델영상과 테스트 영상으로부터 얻어진 특징 벡터간의 거리를 측정함에 의해 인식할 수 있게 된다. 특징 벡터간의 거리에 의한 모델영상을 평면에 나열하고 인식과정에서 거리를 보간함으로써 두 모델의 중간정도의 형상인식도 가능하다. 또, "Size Function"이라는 특수한 방법으로 추출한 특징벡터를 이용하여 손모양으로 표현하는 문자를 인식하는 방법도 널리 알려져 있다[8].

손 형상 연구와 관련된 최근의 연구성과로 그림자를 이용하는 방법[9]과 복수개의 카메라를 사용하는 방법[10]을 들 수 있다. 전자의 경우 간단한 손가락 형상을 대상으로 손 영역과 손에 의해 생긴 그림자의 국소특징정보(peak와 valley)를 이용하여 3개의 손 형상과 관련된 제스처를 인식할 수 있다. 그러나 이 방법은 손과 그림자 투영면과의 거리가 달라지면 특징정보의 매칭이 불안해 안정성에 문제점을 안고 있다. 후자의 경우는 여러 개의 카메라를 사용하여 인식에 필요한 최적의 영상을 선택한 다음 손 영역의 중심점과 손가락의 주방향을 계산하여 8개의 명령을 구별할 수 있는 시스템을 제작하였다. 그러나 이 방법은 손가락의 끝부분이 항상 윗부분을(즉 가상현실 시스템의 디스플레이 방향)을 향하고 있어 환경의 제약을 받는다는 점과 여러 개의 카메라를 사용한다는 점에서 시스템이 커지고 계산량이 많아진다는 단점이 있다.

본 논문에서는 단일 카메라로 들어오는 손 형상을 입력정보로 사용하여 손 형상을 인식한 후 손 형상을 이용하여 수화인식을 위한 손 형상 인식 시스템을 구현하려 한다. 이는 말할 수 있는 비장애인과 말할 수 없는 장애인간의 상호 보다 편리

Collection @ chosun

하고 자연스러운 인터페이스를 제공한다. 비장애인이 장애인과 대화를 시도하려면 먼저 수화로 말하는 방법을 배워야한다는 단점이 있다. 이는 경제적인 면에서도 시간적인 면에서도 비효율적이라고 할 수 있다. 본 논문에서는 이러한 비효율적인 방법을 버리고 비장애인과 장애인간 상호 편리하게 대화할 수 있는 인터페이스를 제공하려 한다. 기존의 연구에서도 비언어적인 수단을 이용하여 정보전달을 하는 방법으로 마우스를 대신하기 위한 얼굴의 표정을 이용하는 방법[11]과 눈의 움직임을 이용하는 방법[12], 눈을 정보전달 장치로 이용하는 방법[13] 그리고 3차원 센서를 이용하는 방법[14]들이 있었다. 얼굴의 표정을 이용하는 방법은 사용자가 충분한 사전학습을 해야 한다는 점에 있어서 사용자와 시스템간의 인터페이스가 자연스럽지 못하다. 또한 눈의 움직임을 이용하는 방법은 눈이 움직일 수 있는 범위 모두를 데이터베이스로 가지고 있어야 함으로 실시간 인식할 때 계산량과 수행시간이 길어지는 단점을 가지고 있으며, 눈만을 직접 사용하는 것이 아니라 눈 주위에 개별적으로 필요한 센서를 부착해야 한다는 불편함을 준다. 한편 3차원 센서를 이용하는 방법 또한 별도의 장비를 구입해야 한다는 점에서 경제적인 부담감을 준다. 또한, 인간의 사고나 인지, 생활의 측면에서 언어의 역할은 절대적이다. 그래서 모든 사람은 언어를 본능적으로 습득해 나간다. 사람은 언어를 습득해 사용하면서 해당 문화나 국가에 대한 정체성을 확립해 나간다. 구어를 통한 언어 습득이 어려운 농아인은 그들만의 독특한 언어인 수화를 통해 사고하고 학습한다[15]. 농아인이 의사소통에 사용하는 수화는 한손 또는 양손을 다양하게 사용해서 자기의 의사를 표현하는 수화와 지화가 있다. 수화는 동적인 손 제스처로 손의 움직임과 손의 모양에 의하여 의미를 전달하며, 지화는 손의 형태에 의하여 한글 자모를 표현하는 지문자의 조합에 의하여 글자를 표현하는 것이다[16]. 다양한 의사 전달을 위하여 농아인은 수화와 지화를 섞어서 사용하며, 한글 표준 수화에서는 명확한 의사 전달을 위하여 조사나 어미 등을 지문자를 이용하여 표현한다. 각 나라마다 자기 나라의 생활 관습에 기초한 표준화된 수화를 정해서 사용하고 있다. 그러므로 농아인의 의사소통 수단으로서 수화의 실시간 인식 시스템이 요구된다. 이렇듯 수화인식의 관심과 중요도가 커지면서 많은 연구자들에 의해 수화인식에 관한 연구가 개발되어졌다. 기존 수화인식에 관한 연구는 손의 움직임 정보를 이용한 방법[17]과, 모바일 환경에서 카메라와 가속도 센서를 사용하는 방법이 있다[18]. 이 외에도 수화인식에 관련된 연구는 현재도 더 좋은 인터페이스를 구현하기 위해 많은 연구자들에

Collection @ chosun

의해 연구가 활발히 진행 중이며 노력을 아끼지 않고 있다.

본 논문에서는 별도의 센서를 부착하지 않은 단일 카메라 환경에서 손 형상을 입력정보로 사용하여 템플리트 매칭에 의한 손가락 끝 점(Fingertip Point)을 인식하여 가상환경을 제어한다. 또한 인식한 손 형상을 신경망 알고리즘을 이용하여 수화인식을 위한 보다 안정적이며 강인한 손 형상 인식 시스템을 구현하고자 한다. 제안하는 방법으로는 카메라로 들어오는 영상을 입력영상으로 하여 피부색 정보를 이용해 배경으로부터 손 영역만을 추출한 후 추출된 손 영역의 형상을 인식한다(전처리과정으로 모델이미지의 사이즈와 압축 및 컬러에 대한 정보를 정규화시켰다). 인식된 손 형상 정보를 템플리트 매칭을 이용하여 실시간적으로 손가락 끝 점을 인식하고[48][49], 인식된 손가락 끝 점만으로 가상환경을 제어한다. 또한 인식된 손 형상 정보에 SOFM(Self Organized Feature Map) 신경망 알고리즘을 적용함으로써 손 형상 인식률에 대한 안전성과 정확성을 향상시켰다.

2. 기존의 주요 손 인식 방법

손 인식 시스템은 일반적으로 다음과 같은 모듈로 이루어져 있다.

- 손 모델링
- 손동작 해석
- 손 형상 및 동작 인식
- 손 형상 기반의 시스템과 응용

손 제스처 모델링은 인식할 제스처의 수학적 모델을 만드는 것이다. 모델링을 하는데 사용된 접근법들은 손 제스처 인식의 중추적 역할을 하고 제스처 해석의 성능에 영향을 준다. 모델 구성 후 비디오 입력으로부터 파라미터를 추출하고 계산을 통해 손 제스처를 인식한다[그림 1-1].

손 형상을 인식하기 위해서는 크게 두 가지 방법으로 분류할 수 있는데 첫 번째로 입력 영상의 배경으로부터 손 영역만을 검출하며, 두 번째로 검출된 손 영역을 손 형상으로 인식하여야 한다. 전자의 경우 배경으로부터 손 영역만을 검출하는 방법으로 물리적인 센서 정보를 이용하는 방법[3], 비디오카메라를 이용하는 방법, Color 정보를 이용하는 방법[18-19], 광류(Optical Flow)[21], 차(Difference)영상[22]

Collection @ chosun

과 같은 프레임간의 차를 이용한 방법, 모델링을 이용하는 방법[23], 컬러 정보와 동영상을 복합해서 사용하는 방법[22,23], 히스토그램과 특징 벡터를 이용한 방법[7] 등이 있으며, 후자의 경우 손 형상을 인식하는 방법으로는 무게중심 프로필(Centroidal Profile) 방법[27], 움직임 방향 정보를 이용한 방법[17], 다중마커를(Multi-Marker) 이용한 방법[28], 상위도 패턴을 이용한 방법[29], 고속카메라와 가속도 센서를 이용한 방법[18] 등이 있다.

[그림 1-1] 손 제스처의 해석과 인식

2-1. 손 영역 검출 관련 연구

가. 물리적인 센서 정보를 이용하는 방법

물리적인 센서를 이용하는 방법 중 가장 일반적인 것은 형상입력 장치인 Data Glove를 사용하는 것이다[3]. [그림 1-2]의 방법은 손가락을 구부릴 때 굴절에 의해 변화하는 빛의 양을 감지하여 구부린 각도를 측정하는 것으로, 측정이 쉽고 자기 센서와 조합하여 사용하면 손의 방향 및 대략적인 3차원 위치를 알 수 있으므로 현재 가상현실 시스템과 같은 인터엑티브한 정보 시스템의 입력 장치로 가장 많이 쓰이고 있다. 그러나 이 방법은 손가락의 자세한 형상을 인식하기 위해서는 아직 정밀도가 낮고 전체 시스템과 접속하기 위해서는 반드시 연결선이 필요하다는 점에서 자연스러운 인터페이스 구축에는 많은 문제점을 안고 있다.

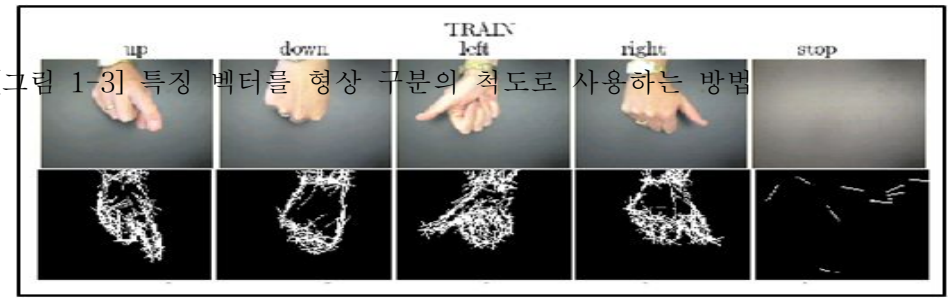
Collection @ chosun

[그림 1-2] 데이터 글로브 종류

나. 비디오카메라를 이용하는 방법

손에 별도의 센서를 장착하지 않고 손 형상을 인식하기 위해서는 시각적인 방법을 이용하는 수 밖에 없다. 그러나 시각적인 방법은 모델구축의 어려움과 세그멘테이션의 어려움 때문에 알기 쉬운 표식을 붙이거나[5] 손의 형상이 뚜렷이 분리될 수 있는 간단한 문제를 위주로 적용되어 왔다. 또 난이도와 해상도의 문제 때문에 손가락의 형상의 자세한 인식 보다는 상반신을 포함한 어깨 전체의 제스처 인식을 연구 목표로 삼고 있는 경우가 많다[6]. 손 형상의 인식의 대표적인 방법은 화소간의 휘도 경도를 구해 히스토그램을 만들고 이 히스토그램을 평활화 하여 얻어진 특징 벡터를 형상 구분의 척도로 삼는 방법이 있다[7][47]. 즉 모델영상과 테스트 영상으로부터 얻어진 특징 벡터간의 거리 측정에 의해서 인식할 수 있게 된다.

특징 벡터간의 거리에 의한 모델영상을 평면에 나열하고 인식 과정에서 거리를 보간 함으로써 두 모델의 중간정도의 형상 인식도 가능하다. 또, “Size Function”이라는 특수한 방법으로 추출한 특징벡터를 이용하여 손 모양으로 표현하는 문자인식 방법도 널리 알려져 있다[그림 1-3].



[그림 1-3] 특징 벡터를 형상 구분의 척도로 사용하는 방법

Collection @ chosun

손 형상 연구와 관련된 최근의 연구 성과로 그림자를 이용하는 방법[9]과 복수 개의 카메라를 사용하는 방법[26]을 들 수 있다. 전자의 경우 간단한 손가락 형상을 대상으로 손 영역과 손에 의해 생긴 그림자의 국소 특징정보(peak 와 valley)를 이용하여 3개의 손 형상과 관련된 제스처를 인식할 수 있다. 그러나 이 방법은 손과 그림자 투영면과의 거리가 달라지면 특징정보의 매칭이 불안해 안정성에 문제점을 안고 있다. 후자의 경우는 여러 개의 카메라를 사용하여 인식에 필요한 최적의 영상을 선택한 다음 손 영역의 중심점과 손가락의 주방향을 계산하여 8개의 명령을 구별할 수 있는 시스템을 제작하였다. 그러나 이 방법은 손가락의 끝부분이 항상 영상의 윗부분(즉 가상현실 시스템의 디스플레이 방향)을 향하고 있어 환경의 제약을 받는다는 점과 여러 개의 카메라를 사용한다는 점에서 시스템이 커지고 계산량이 많아진다는 단점이 있다.

카메라를 통하여 들어 온 영상만을 근거로 손의 형상을 인식하는 데는 크게 두 가지의 문제점이 따른다. 첫째, 손은 인체 중에서 가장 많은 자유도를 지닌 부위로 평상시 각 구성체인 손가락이 서로 겹치거나 보는 시각에 따라 거의 무한 종류의 형상을 지니고 있어 인식을 위해 필수적인 세그멘테이션이 매우 어렵다는 점이다. 즉 손의 특정 형상에 안정적으로 대응되는 에지나 코너 등의 기하학적 특징 추출이 어렵게 된다. 이 때문에 정확한 형상을 얻기 위해서는 불편하더라도 물리적인 센서를 직접 부착을 하거나, 손가락 간에 서로 겹치는 현상이 일어나지 않는 제한적인 모양만을 인식의 대상으로 삼는 수 밖에 없다.

둘째, 손은 크기에 비해 움직이는 범위가 매우 넓어 적절한 해상도의 영상을 얻기가 어렵다는 점이다. 즉, 형상을 인식하기 위해서는 카메라를 매우 가까이 접근시켜서 촬영하거나 렌즈를 사용하여 충분한 크기의 영상을 확대할 필요가 있는데 그 크기에 비해 상대적으로 손이 움직이는 속도가 매우 빠르고 범위가 넓기 때문에 움직임 정보를 유효하게 이용하기가 어렵게 된다. 따라서 수화인식과 같은 실제 응용에서는 원거리의 카메라를 이용하여 손의 대략적인 위치를 알아내는 과정과 손의 위치에 초점거리가 짧은 카메라의 시선을 맞추어 그 형상을 인식하는, 즉 해상도가 다른 두 가지 인식과정이 필요하게 된다.

다. Color 정보를 이용한 방법

HSI, YIQ, 정규화된 RGB 컬러 모델 등에 의해 살색과 유사한 색을 영상에서 찾

Collection @ chosun

아 손 영역을 추출한다. 그러나 이 방법은 보통의 그레이 영상보다 3배 이상의 데이터가 사용되어 처리량이 증가하게 된다[그림 1-4].

(a) Random한 배경에서의 손 형상 모델

[그림 1-4] Color 정보에 의한 손 영역 추출

(b) 모델 영상에 대한 손 영역 추출

라. 광류(Optical Flow), 차 영상 등 프레임간의 차를 이용한 방법

그레이 영상만을 이용하므로 처리해야하는 데이터양이 적으나 움직임이 없는 경우 손 영역을 검출하기 어려운 단점이 있다. [표 1-1]은 Optical Flow 알고리즘이다. [그림 1-5], [그림 1-6], [그림 1-7]까지는 광류에 의한 손 영역 추출 방법이며, [그림 1-8]은 차 영상에 의한 손 영역 추출 방법이다.

(1) 광류에 의한 손 영역 추출 방법

[표 1-1] Horn의 Optical Flow 계산 알고리즘

1. 모든 (i,j)에 대해서 속도 벡터 를 초기화한다.
2. 를 반복 횟수라 하고, 모든 픽셀 (i,j)에 대해서 다음 식을 사용하여 u^k, v^k를 계산한다.

$$c(i,j)$$

k

$$u^k(i,j) = \overline{u^{k-1}} - I_x(i,j)\frac{P(i,j)}{D(i,j)}$$

$$v^k(i,j) = \overline{v^{k-1}} - I_y(i,j)\frac{P(i,j)}{D(i,j)}$$

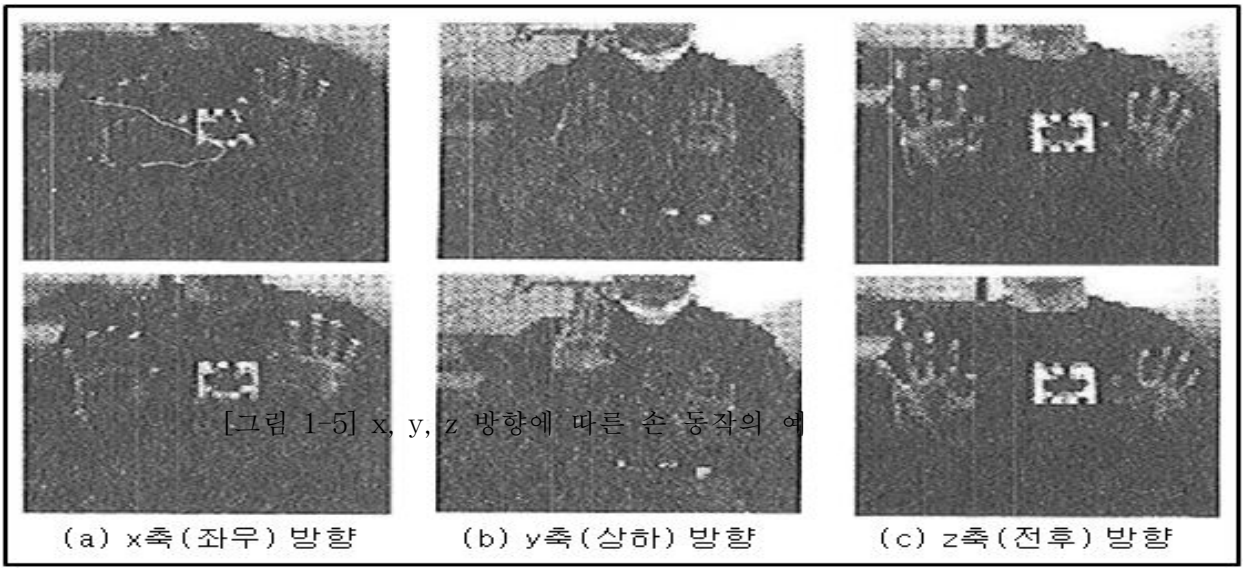
Collection @ chosun

여기서 $P=I_x u+I_y v+I_t, D=\lambda^2+I_x^2+I_y^2$를 나타낸다.

3. 다음 조건을 만족하면 계산을 멈추고, 아니면 과정 2를 반복한다.

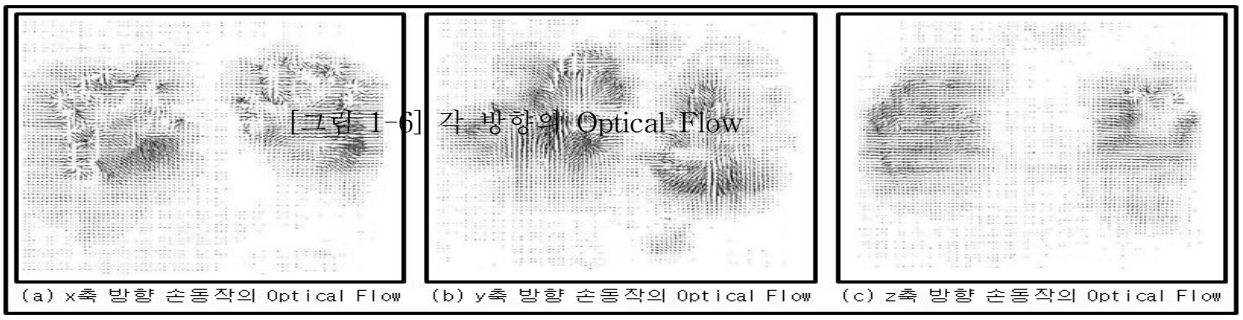
$$\sum_i \sum_j E^2(i,j) < \varepsilon$$

여기서 ε 은 최대 허용 오차를 나타낸다.



(a) x축(좌우) 방향 (b) y축(상하) 방향 (c) z축(전후) 방향

[그림 1-5] x, y, z 방향에 따른 손 동작의 예



(a) x축 방향 손동작의 Optical Flow (b) y축 방향 손동작의 Optical Flow (c) z축 방향 손동작의 Optical Flow

[그림 1-6] 각 방향의 Optical Flow

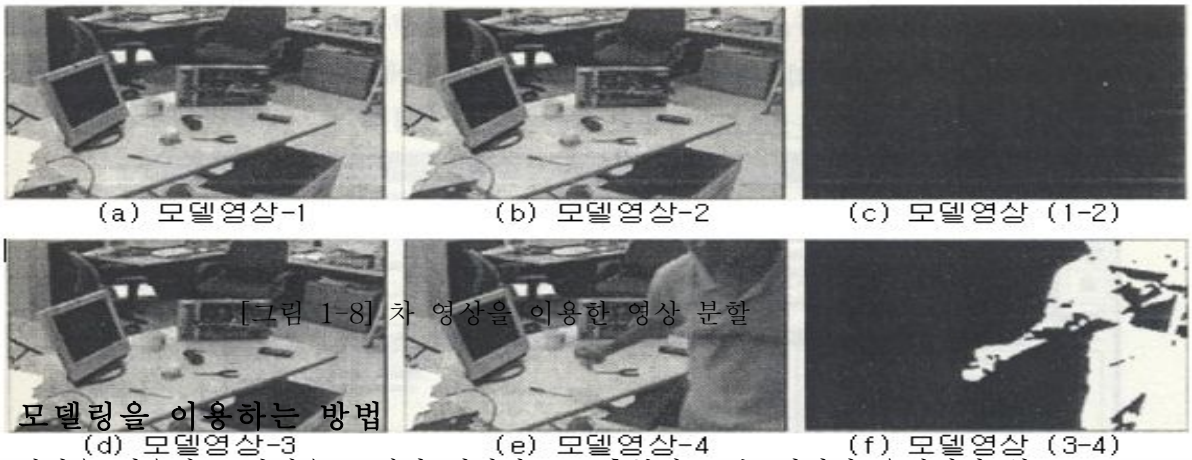
(a) x축 방향 손동작의 영역 추출 (b) y축 방향 손동작의 영역 추출 (c) z축 방향 손동작의 영역 추출

[그림 1-7] 각 방향의 손 영역 추출 결과

Collection @ chosun

(2) 차 영상에 의한 손 영역 추출 방법

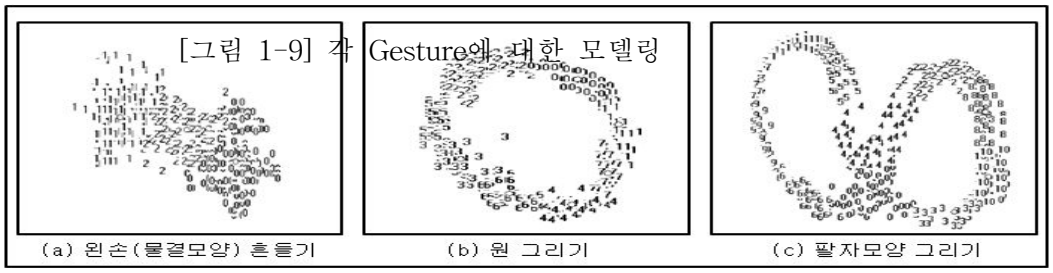
두 영상의 차이를 이용하는 방법은 보안 시스템이나 영상 분석 시스템 등 여러 응용에서 사용될 수 있다. 보안 시스템에서는 두 영상의 차이를 구함으로써 침입자의 출현을 감지할 수 있다. 예를 들어, 야간에 아무런 물체의 움직임도 없어야 할 장소를 카메라로 계속 촬영하면서 연속된 두 영상사이의 차이를 구한다고 가정하자. [그림 1-8]의 (a),(b)와 같이 연속된 영상에서 침입자가 없는 경우에는 두 영상 사이에 아무런 차이가 없으므로 두 영상의 뺄셈 영상에서 (c)와 같이 모든 픽셀의 값이 "0"에 가까울 것이고 (e)와 같이 침입자가 나타날 경우에는 두 영상의 뺄셈 영성에서 (f)와 같이 침입자가 나타난 부분의 픽셀값들이 "0"보다 훨씬 큰 값을 가질 것이다.



[그림 1-8] 차 영상을 이용한 영상 분할

마. 모델링을 이용하는 방법

모델링을 이용하는 방법은 그레이 영상만으로 충분하고 손 영역의 움직임이 없어도 손 영역을 검출할 수 있으나 손 모양에 따라 여러 가지 모델이 필요하며 처리 시간이 길어지는 단점이 있다[그림 1-9].

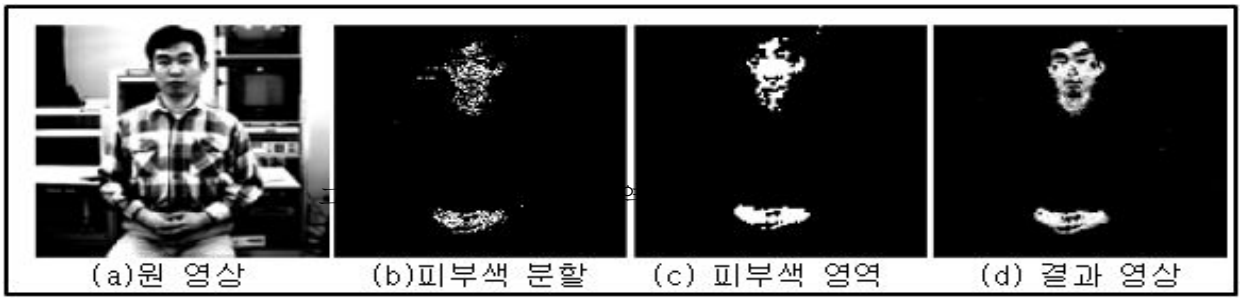


[그림 1-9] 각 Gesture에 대한 모델링

Collection @ chosun

바. 컬러 정보와 동영상을 복합해서 사용하는 방법

컬러 정보와 동영상을 복합해서 사용하는 방법은 위의 세 가지 방법(컬러영상을 이용하는 방법, 광류·차 영상을 이용하는 방법, 모델링을 이용하는 방법)을 복합적으로 사용하는 방법이다. [그림 1-11]은 전처리 과정에 의해 추출된 얼굴영역과 손 영역에 대한 히스토그램이다.



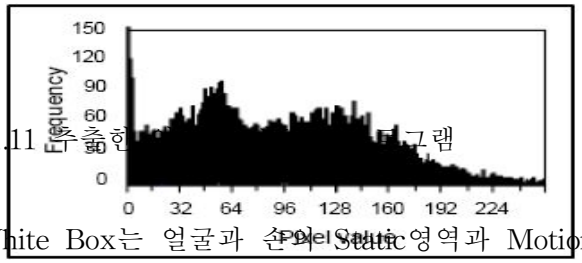
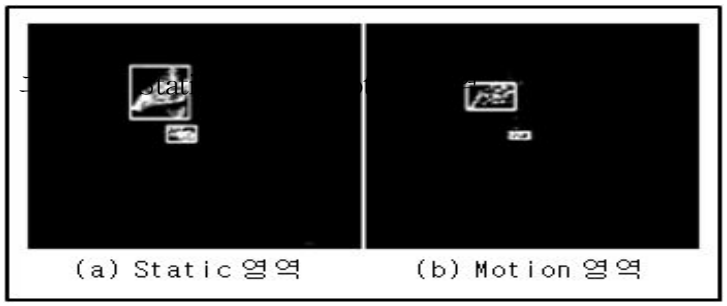


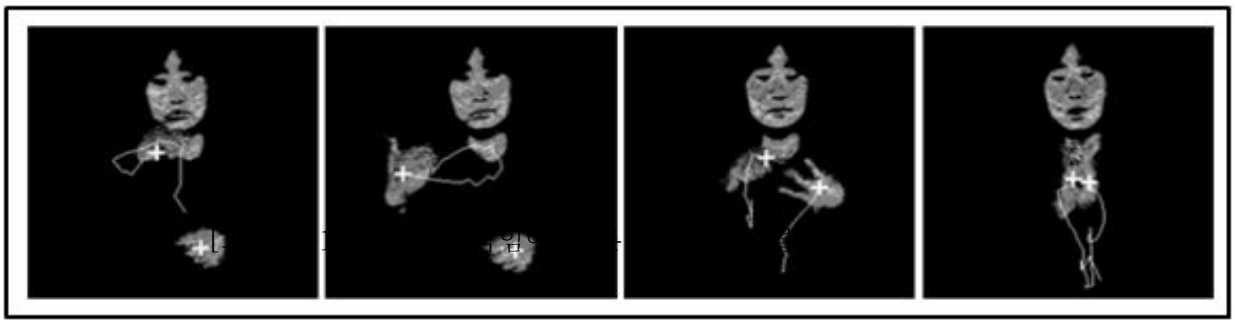
그림 1.11 추출한 [illegible] 그램

[그림 1-12]에 만들어진 White Box는 얼굴과 손의 Static영역과 Motion영역을 나타낸다. Static 영역은 얼굴과 손이 움직이지 않는 상태를 말하며, Motion 영역은 얼굴과 손이 고정된 상태가 아닌 움직임이 있는 상태를 말한다. 즉, 얼굴과 손의 움직임에 따른 Tracking Blob을 말한다.



Collection @ chosun

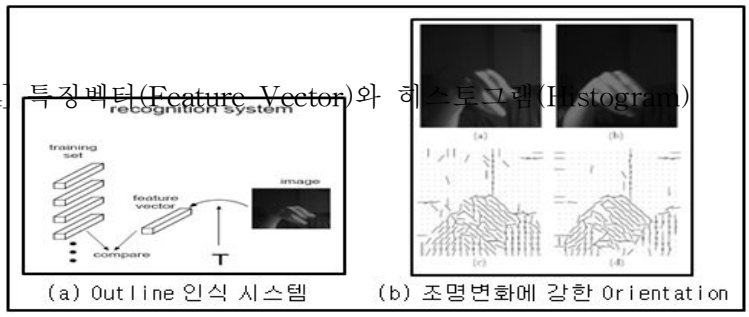
[그림 1-13]은 손의 움직임에 따른 Tracking 과정이며, 손 인식 결과는 “+”모양을 나타내고 있다.



사. 히스토그램과 특징 벡터를 이용한 방법

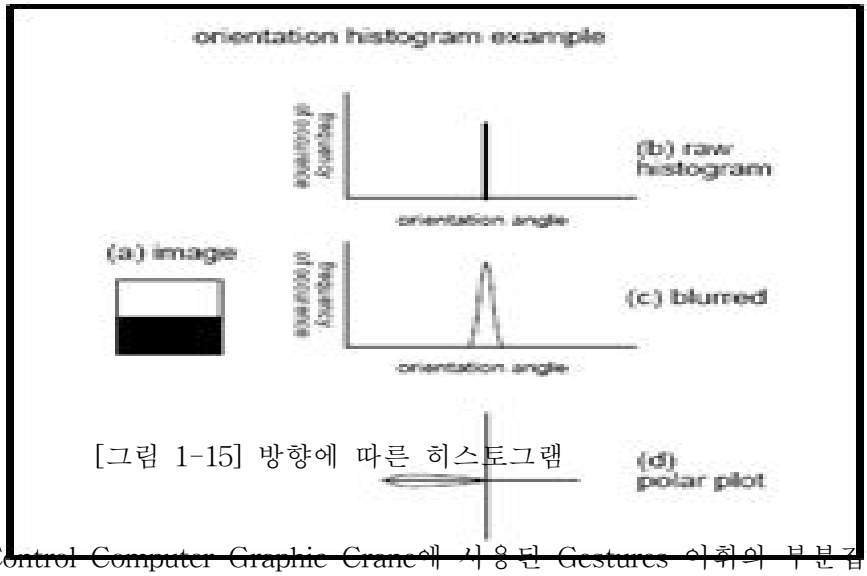
손 형상의 인식의 대표적인 방법은 화소간의 휘도, 경도를 구해 히스토그램을 만들고 이 히스토그램을 평활화하여 얻어진 특징 벡터를 형상 구분의 척도로 삼는 방법이 있다[7]. [그림 1-14](a)는 영상의 Outline을 인식하는 시스템으로서 Feature Vector를 구하여 영상 데이터를 변환하여 Particular Gesture를 얻는다. 그리고 사전에 산출된 Training Set으로부터 Feature Vector를 비교한 후 연속적인 Input Image의 Dynamic Gesture를 인식한다. [그림 1-14](b)는 방향정보를 이용하여 지역적인 조명변화에도 강인한 모습을 보여준다. Pixel의 농도는 조명변화에 민감하다. [그림 1-14](b)에서 (a)와 (b)는 두 가지 다른 조명 상태 하에서 Hand Gesture를 설명하고 있으며 이것은 조명변화에 따라 Pixel 농도 변화 과정을 나타낸다. (c)와 (d)는 방향정보를 이용하면 지역적인 조명변화에도 안정된 결과를 보여준다는 것을 나타낸다.

[그림 1-14] 특징벡터(Feature Vector)와 히스토그램(Histogram)



Collection @ chosun

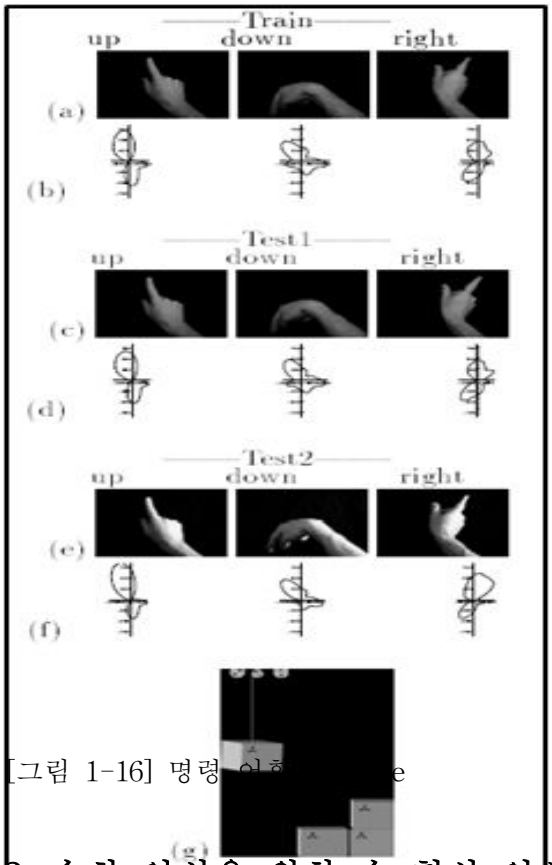
[그림 1-15]는 방향성 히스토그램을 설명하고 있으며, (a)는 하나의 높은 Contrast를 갖는 Orientation을 나타내고, (b)는 하나의 방향만 갖는 히스토그램((a)의 히스토그램)을 나타낸다. (c)는 (a)의 경계선 지점에서의 이웃한 화소에 따른 히스토그램을 나타내며, (d)는 끝지점(극지점)에서의 상태를 나타낸다.



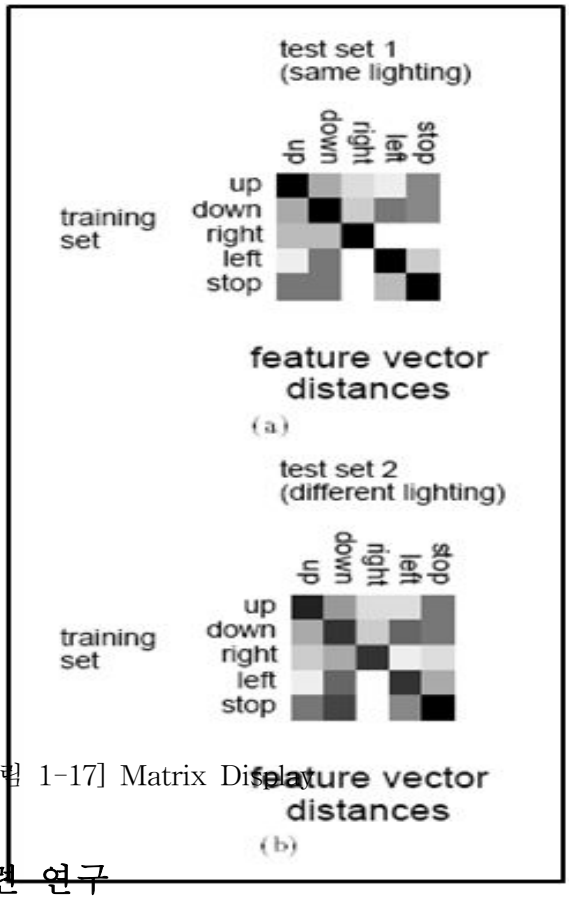
[그림 1-15] 방향에 따른 히스토그램

[그림 1-16]은 Control Computer Graphic Crane에 사용된 Gestures 어휘의 부분집합들이다. (a)는 up, down and right의 명령을 위한 Gesture의 Training Set이고, (c)는 Test Set을 위해 (a)와 같은 조명상태 하에서 같은 Gesture를 표현했으며, (e)는 다른 조명상태 하에서 만든 Test Set이다. (b)와(d) 그리고 (f)는 각각 부합하는 방향성 히스토그램이다. (g)는 외부에서 어휘 명령들을 실시간으로 Computer Graphic Crane에서 제어할 수 있다는 것을 보여주고 있다. [그림 1-17]은 Gray Level 상에서 Training set과 Test set 사이의 거리를 Matrix로 나타내고 있다. (a)는 같은 조명하에서 Test Gesture를 만들었다. 이것은 매우 잘 분류되었고 Training Gesture에 일치하며, Matrix의 특징 벡터의 거리를 어두운 대각선으로 가리켰다. (b)는 다른 조명상태 하에서 Test Gesture를 만든 경우이며, 밝은 조명환경에서 같은 정도의 지시를 하고 정확한 답을 줄 때까지 가장 인접한 분류한다.

Collection @ chosun



[그림 1-16] 명령 인식 [illegible]e



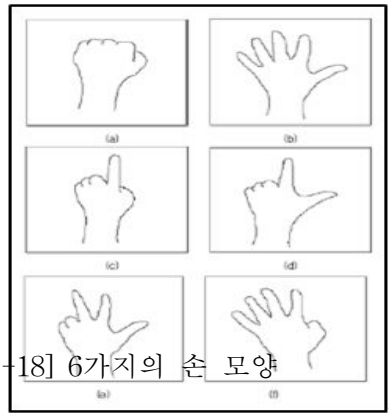
[그림 1-17] Matrix Display

2-2. 수화 인식을 위한 손 형상 인식 관련 연구

가. 무게중심 프로필(Centroidal Profile) 방법

수화를 인식하기 위해 손 형상을 인식함에 있어 전처리 과정으로 손 영역을 추출한 후 그 영역에 대한 윤곽선을 검출하고 개선된 무게중심 프로필(Centroidal Profile) 방법을 사용한다. 서로 다른 6가지 모양의 수화를 대상으로 인식을 수행했으며 모양은[그림 1-18]과 같다. [그림 1-18]의 6가지 손 모양은 한국 수화에서 사용하는 것 중에서 몇 가지를 선택한 것이다. (a)는 알파벳 S와 익다(열매맺다), (b)는 닮다(모방하다), (c)는 한글 모음 “ㅏ”와 “내일”을 각각 나타내는데 사용한다. (d)는 알파벳 L, 아니요(부정)와 해(태양), (e)는 무지개와 배구, (f)는 한글 자음 “ㅂ”과 “대학교”를 표시하는데 각각 사용한다. 같은 손 모양이 상황에 따라 알파벳, 동사, 한글 자음, 한글 모음, 명사 등으로 사용된다.

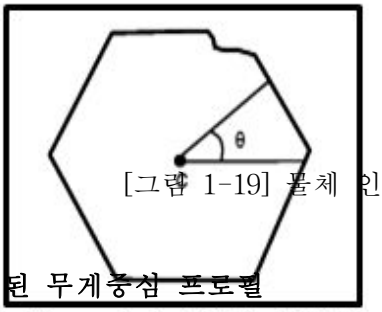
Collection @ chosun

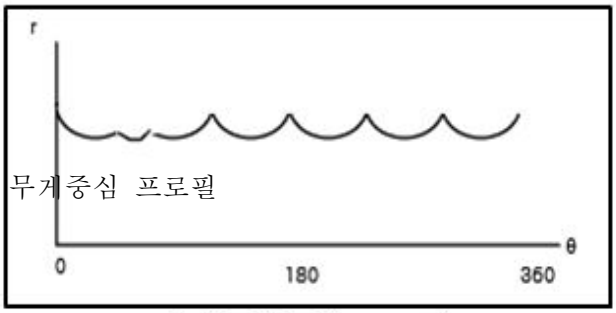


[그림 1-18] 6가지의 손 모양

(1) 무게중심 프로필(Centroidal Profile)

무게중심 프로필[30][그림 1-19](a)는 검출된 물체 영역에 대해 물체의 외곽을 각도로 표현함으로써 물체의 형태를 판단하는데 사용된다. 대부분 이러한 형태의 알고리즘은 물체의 무게중심을 구하고 무게중심으로부터 각도에 따른 거리(r) 값으로 표시한다. [그림1-19](b)는 물체 인식을 위한 무게중심 프로필을 보여준다. 물체의 형태는 무게중심 프로필을 구한 후 이상적인 물체에 대한 프로필과 비교하여 파악된다.





[그림 1-19] 물체 인식을 위한 무게중심 프로필

(2) 수정된 무게중심 프로필

물체의 외곽선이 [그림 1-20]와 같은 모양의 경우에는 한 각도에 대해 무게중심으로부터 두 개의 가지로 정합시 문제가 된다. 이러한 문제를 해결하기 위한 방법으로 주어진 각도( )에 대해 무게중심으로부터 제일 가까운 값을 취한다. 그러나 손의 경우는 이러

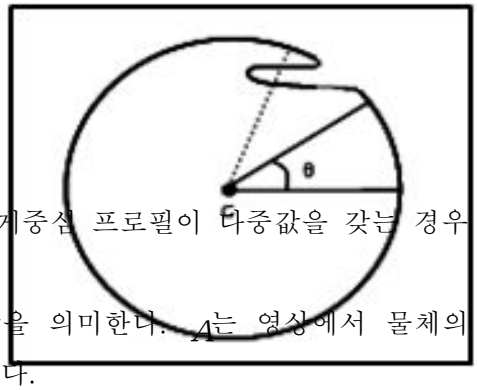
θ

Collection @ chosun

한 형태로 표현할 수 없다. 따라서 각도를 사용하지 않고 윤곽선 상에서 모든 화소들을 대상으로 한 무게중심 프로필을 제안하다. 윤곽선 검출을 위해 8방향 코드를 사용하였으며, 윤곽선 위에 존재하는 모든 점들(x_i, y_i)에 대해 무게중심 ($\bar{x}, \bar{y}$)으로부터 거리(r)를 (식 1-1)에 의해 구한다. 2진 영상에서 각 화소에 대한 밝기값(Intensity)을 그 점에 대한 무게로 간주하면 면적의 중심은 무게중심과 같게 된다. 그러므로 무게중심은 (식 1-2)에 의해 구해진다.

$$r = \sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2} \qquad \text{(식 1-1)}$$

$$\bar{x} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M} jI(i,j)}{A}, \quad \bar{y} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M} iI(i,j)}{A} \qquad \text{(식 1-2)}$$



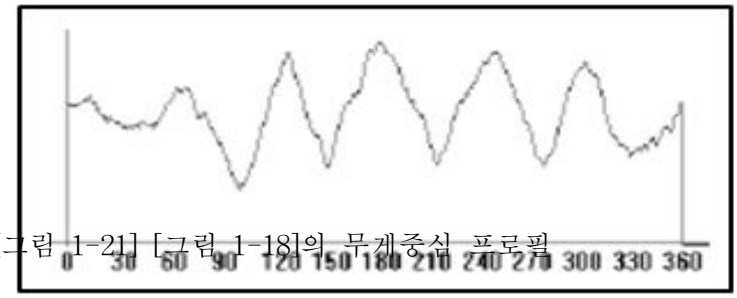
[그림 1-20] 무게중심 프로필이 다중값을 갖는 경우

여기서 $I(i,j)$는 화소의 밝기값을 의미한다. A는 영상에서 물체의 전체 화소수를 뜻하며 (식 1-3)과 같이 표현된다.

$$A = \sum_{i=1}^{N}\sum_{j=1}^{M} I(i,j) \qquad \text{(식 1-3)}$$

Collection @ chosun

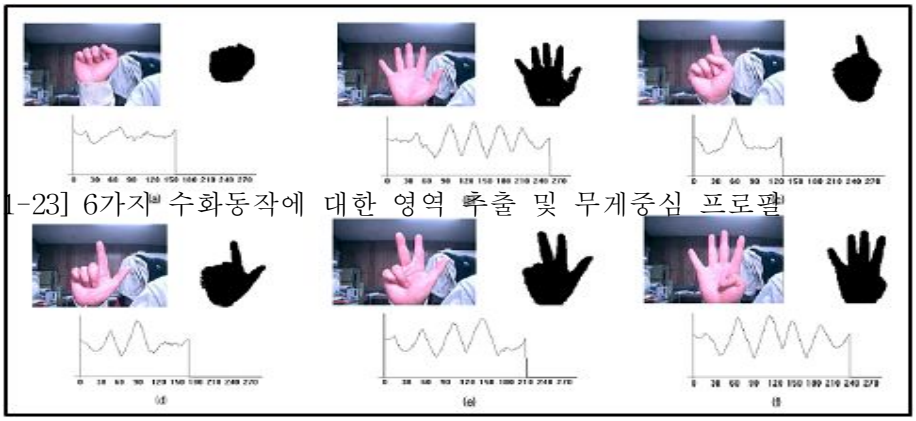
[그림 1-21]은 [그림 1-18](b)의 손 모양에 대해 무게중심 프로필을 구한 것이다.



[그림 1-21] [그림 1-18]의 무게중심 프로필

[그림 1-22]는 손 형상에 대한 윤곽선을 기반으로 무게중심 프로필을 적용하여 수화를 인식하는 과정을 보이며, [그림 1-23]은 6가지의 수화동작에 대해 손 영역을 추출한 후 무게중심 프로필을 구한 결과이다.

[그림 1-22] 무게중심을 이용한 수화인식 시스템



[그림 1-23] 6가지 수화동작에 대한 영역 추출 및 무게중심 프로필

Collection @ chosun

나. 움직임 방향정보를 이용한 방법

수화단어는 간단한 동작에서 복잡한 동작까지 많은 경우가 존재하지만 인식에 사용된 단어들은 [그림 1-24]와 같이 한 손 동작이면서 손의 움직이는 방향만으로 인식할 수 있는 간단한 단어로서 “우리”,“형”,“남동생”,“언니”,“여동생”,“새벽”,“저녁”,“젊다”,“하루” 등의 9가지 단어이다. 선택된 단어들 중에서 “형”과 “언니”,“남동생”,“여동생”은 움직임이 같으므로 인식결과에는 하나의 동작으로 간주한다.

[그림 1-24] 인식에 사용된 수화 단어(9가지)

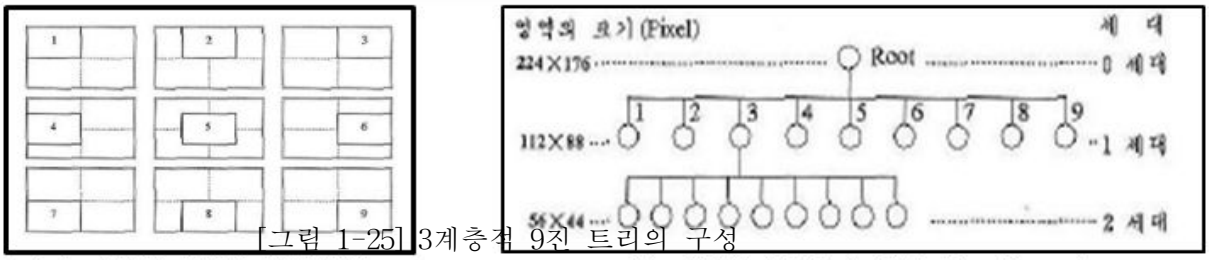
인식을 하기 위해 은닉 마르코프 모델(HMM: Hidden Markov Model)을 사용한다. 인식모델로서 은닉 마르코프 모델은 마르코프 체인을 근거로 하는 시계열패턴의 확률적인 생성모델이며, 동영상과 같이 시간적으로 변화하는 패턴에 대한 인식에 적합한 특성을 지니고 있다. 손의 이동방향 정보를 HMM에 적용하기 위하여 관측열을 인식에 알맞은 형태로 변환한다. 손의 이동방향은 동영상의 매 프레임마다 9진트리[그림 1-25] 2세대 노드 중에서 최대 특징값을 갖는 노드에 대응되는 차영상 영역(손영역)의 이동방향으로 한다. 따라서 손 영역의 크기(56*44)를 이동단위로 하면 이동범위는 7*7의 이동분해 기능을 갖게 된다. (9진트리:차영상을 가로와 세로 각각 1/2 크기의 소영역 9개로 중첩하면서 분할하고, 그 소영역을 다시 9개로 반복 분할하여 영역을 계층적으로 9진트리에 대응시키면서 각 노드마다 특징값을 할당하는 방법이다.)

검출된 프레임마다의 방향을 [그림 1-26](a)의 8방향의 체인코드로 변화하여 HMM의 입력으로 한다. HMM은 Left-to-Right 모델로 하고, 파라메터 $=\{N, M, A, B, \Pi\}$에 대해서 [그림 1-26](b)와 같이 상태의 수 $N=4$, 관측심볼의 수는 체이코드의 종류로 하여 $M=8$, 초기상태 확률 분포 Π는 등확률로 설정한다.

λ

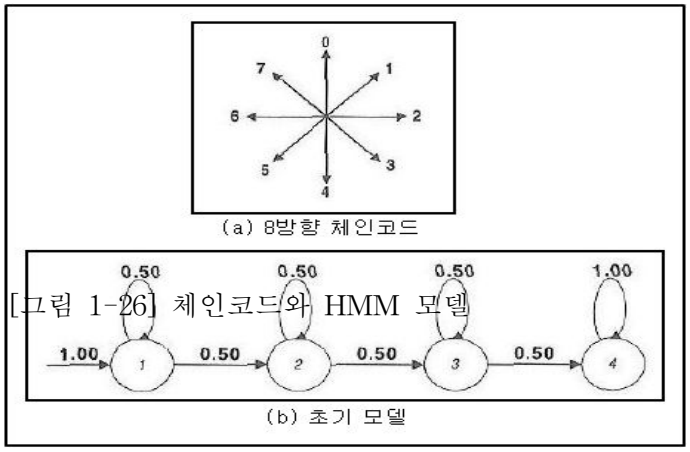
Collection @ chosun

위와 같이 정의된 HMM 모델의 파라메터는 Baum Welch 방법을 이용하여 학습하였다. [그림 1-27](a)에 “남동생”이란 단어를 5사람이 2번 동작한 10개 데이터로 학습한 상태전이 확률분포 A와 관측심볼 확률분포 B를 예시했다.



(a) 차영상의 영역 분할 방법 (b) 차영상의 영역에 대응되는 9진 트리

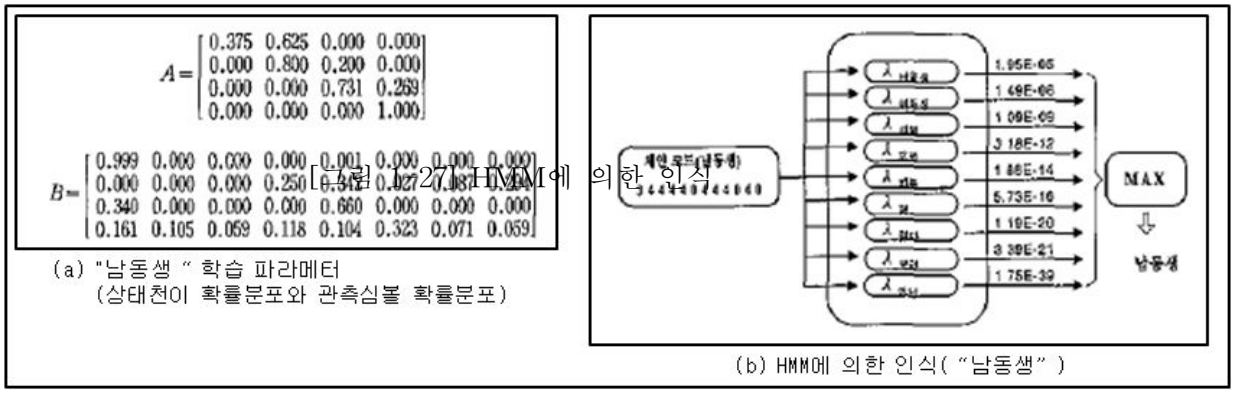
[그림 1-25] 3계층적 9진 트리의 구성



(a) 8방향 체인코드

(b) 초기 모델

[그림 1-26] 체인코드와 HMM 모델



(a) "남동생 " 학습 파라메터
(상태천이 확률분포와 관측심볼 확률분포)

(b) HMM에 의한 인식(“남동생”)

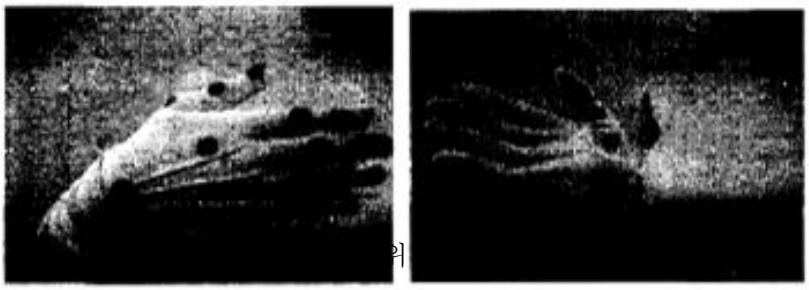
[그림 1-27] HMM에 의한 인식

Collection @ chosun

다. 다중마커(Multi-Marker)를 이용한 방법[28]

(1) 마커의 위치 설정

마커를 많이 설정할수록 각 손가락의 미세한 움직임도 알 수 있지만 동일마커를 인식하기는 어렵다. [그림 1-28]은 손의 움직임을 알아내기 위해 손의 관절 부분에 마커를 부착하여 손의 이동과 회전정도를 손 앞면에는 16개, 손바닥에는 4개의 마커를 정했다.



(a) 손 앞면 마커 위치 (b) 손 바닥 마커 위치

손 앞면의 경우 손(위) 굽힘 정(Flection) 등의 정보를 얻기 위해 손가락의 두 번째와 세 번째 마디에 모두 마커를 부착했으며, 손바닥의 경우 굽힘 운동(Flection)에 따라 앞면의 마커를 통해 손의 모양 등을 파악할 수 있다. 앞뒤 손목 부위의 마커는 가장 큰 마커로 손의 이동을 파악하는데 구별이 되도록 만들었다.

(2) 마커의 인식 절차

① CCD 카메라로 촬영한 영상을 2진화 영상으로 만든다.

② [그림 1-29](a)의 2진화 영상에서 (b)와 같이 스캔하여 읽다가 세 픽셀 이상이 주위보다 검은색으로 계속되면 마커로 인식한다.

③ A점과 같이 마커 내부의 한 점이 발견되면 이로부터 세로로 증가하여 마커의 경계점인 B점을 찾는다.

④ 선분 의 중점 P를 찾는다.

⑤ P점에서 마커의 가로 양끝지점 C, D를 찾는다.

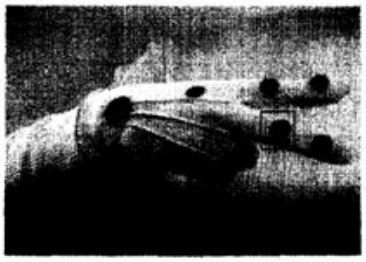
⑥ 선분 의 중점 Q를 찾는다.

⑦ 찾은 중심값을 비교하여 변화가 적으면 마커의 중점 R을 결정한다.

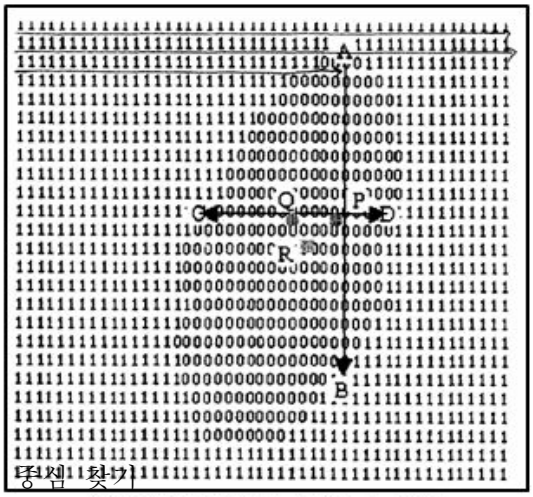
$\overrightarrow{AB}$

$\overrightarrow{CD}$

Collection @ chosun



(a) "가위 "를 뜻하는 수화

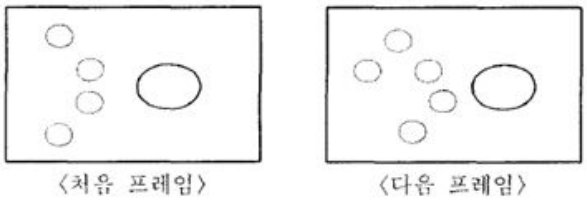


(b) 마커의 인식 및 중심 찾기

[그림 1-29] 마커의 중심 찾기

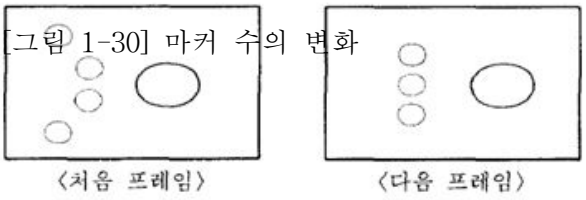
(3) 마커의 추적

마커 인식을 한 후 전 프레임의 마커 위치 정보를 이용하여 다음 프레임의 마커 중 동일 마커를 찾아낸다. 이 때, 손의 모양과 위치, 운동 속도에 따라서 마커의 수가 줄어들기도 하고 늘어나기도 함으로 마커가 보이기도 하고 사라지기도 한다. 이런 이유로 여러 개의 마커를 단순히 거리상으로 가장 최소로 움직인 것을 자신의 마커로 인식하면 안 된다.



〈처음 프레임〉 〈다음 프레임〉

(a) 마커의 수가 늘어난 경우



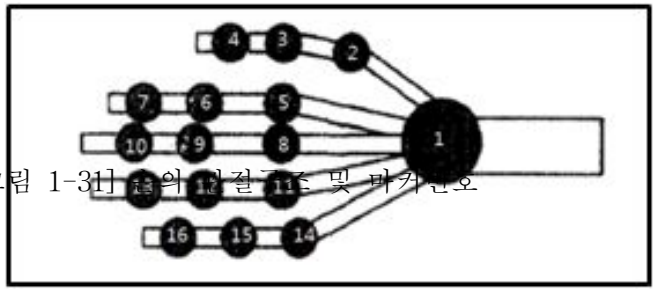
〈처음 프레임〉 〈다음 프레임〉

(b) 마커의 수가 줄어든 경우

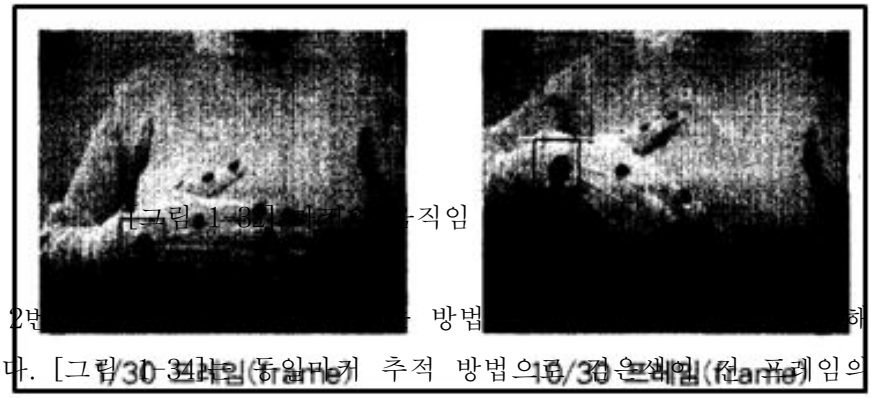
[그림 1-30] 마커 수의 변화

Collection @ chosun

영상에서 마커를 추적할 때 손의 이동량과 회전량을 계산하고 마커의 위치를 예측하여 동일마커를 찾는다. 손목부위(1번) 마커의 움직임량을 통해서 손의 이동량을, (1번,2번)마커의 벡터량을 통해서 손의 회전량을 계산한다. [그림 1-31]은 이동과 회전에 의한 움직임을 분리하기 위해 마커들에 고유번호를 붙인 것이다. 마커가 순서적으로 연결되어 있으므로 연속동작에서 마커를 추적할 때 손목부위의 마커 움직임을 추적한 후 각각의 손가락의 마커 움직임을 알아낸다. [그림 1-32]는 1/30 프레임과 10/30 프레임 사이의 손의 움직임을 보여준다. 손의 움직임은 이동과 회전으로 이루어지는데, 손목부위의 가장 큰 마커의 이동을 통해 파악되며 얻어진 마커의 이동량은 나머지 마커에도 적용된다.



[그림 1-31] 손의 마디절점 및 마커번호



[그림 1-32] 마커의 움직임

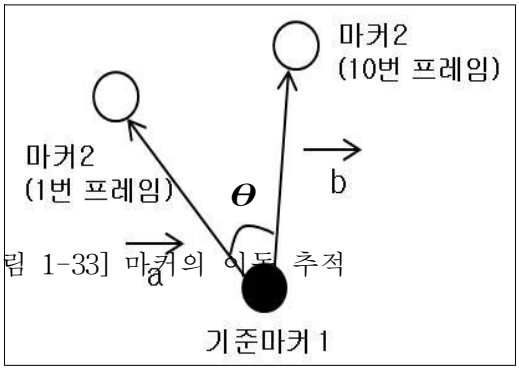
[그림 1-33]은 2번 [illegible] 방법 [illegible]하 손의 회전을 안다. [그림 1-34]는 동일마커 추적 방법으로 검은색이 전 프레임의 마커이고 회색이 다음 프레임의 마커이다. 기본 식은 (식 1-4),(식 1-5)와 같다.

Collection @ chosun

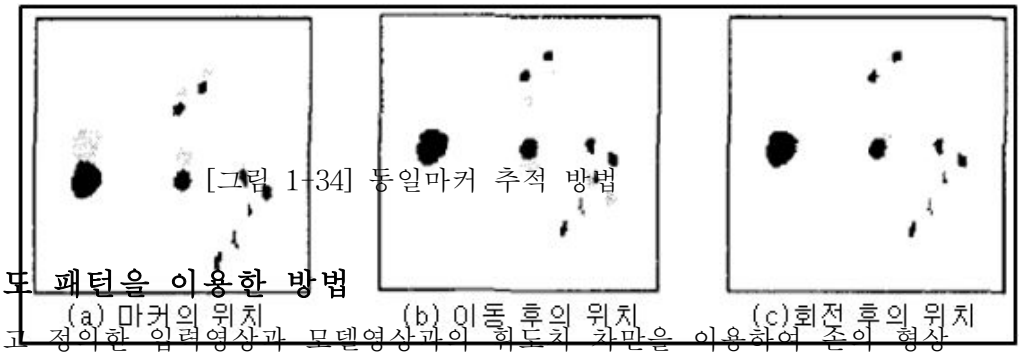
$$a \cdot b = |a||b|\cos\theta \qquad \text{(식 1-4)}$$

$$\cos\theta = a \cdot b / |a||b| \qquad \text{(식 1-5)}$$

(식 1-4),(식 1-5)에서 얻어진 θ 값을 그림의 마커번호 3,4,5번에도 적용하여 움직임을 추적한다.



[그림 1-33] 마커의 이동 추적



[그림 1-34] 동일마커 추적 방법

라. 상위도 패턴을 이용한 방법

상위도라고 정의한 입력영상과 모델영상과의 휘도치 차만을 이용하여 좋은 형상을 인식한다. 휘도치의 차이란 화소간의 뺄셈에 의하여 구해지므로 결국 두 영상간의 상관관계를 의미한다. (식 1-6)에 의해 구해지는 값은 두 영상의 서로 다른 정도를 나타내므로 “상위도”라는 용어를 특별히 정의하여 사용한다.

Collection @ chosun

(1) 상위도 공간의 정의

(식 1-6)은 영상들 사이의 상위도를 정의하는 식이다.

$$D_{i,j} = \underset{(p,q)\in R_l}{Min}\left\{\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\left|I_i(p+m,q+n)-I_j(m,n)\right|\right\} \quad \text{(식 1-6)}$$

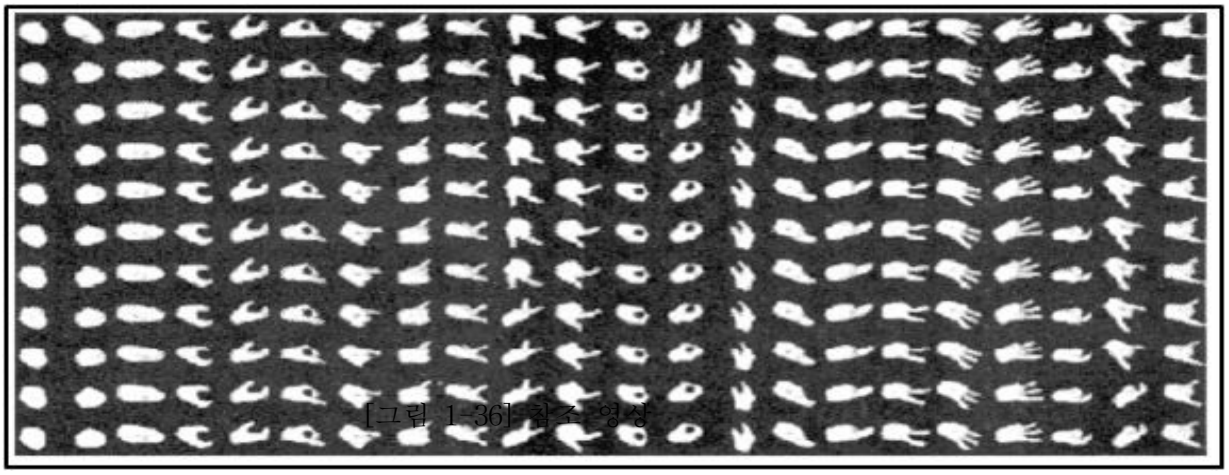
위 식은 임의의 영상 I_i를 다른 영상 I_j와 영상영역 R_l내에서 가장 닮은 위치인 좌표 (p,q)에 겹쳤을 때 서로 겹쳐진 화소가 갖는 휘도치 차이의 절대값을 겹쳐진 영역(N * M 범위)에 걸쳐 구한 최소값이다. $D_{i,j}$의 값은 화소간의 뺄셈에 의해 구해지므로 넓은 의미에서 두 영상의 상관관계(특히 영상이 급격히 변하지 않는 경우)를 나타낸다고 볼 수 있으나 엄밀히 보아 단순히 두 영상 I_i와 I_j가 서로 산술적으로 얼마나 다른가를 나타내므로 “상위도”라는 용어로 사용한다.

(2) 상위도 공간의 생성

손 형상을 인식하기 위해 25종류의 영문 알파벳 모양의 손 형상을 모델영상으로 갖는다. [그림 1-35]는 25종류의 모델영상이며 [그림 1-36]은 참조영상으로서 손과 카메라가 일정거리를 유지하고 손목을 광축을 기준으로 (-45)에서 (+45)의 범위에서 자연스럽게 회전시킨 영상이다.

[그림 1-35] 미국 수화의 표준으로 사용되는 알파벳 손가락문자(a~y) 영상

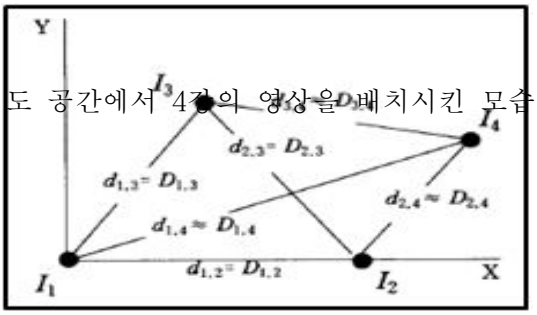
Collection @ chosun



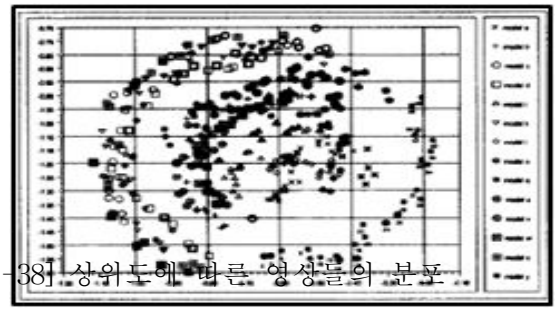
[그림 1-36] 참조 영상

상위도 공간이란 상위도를 척도로 입력영상의 위치를 규정하는 2차원 직교공간을 말한다. 이 공간내에서 각 영상들은 상호간에 상위도 만큼의 거리를 두고 떨어져 놓이게 된다. 따라서 서로 비슷한 영상은 상위도가 적으므로 근접한 위치에 놓여 하나의 집단을 이루게 된다. [그림 1-37]은 4장의 영상($I_1 \sim I_4$)이 순서대로 이 공간에 놓이는 모습을 보여준다. 먼저 영상 I_1을 이 공간의 원점 $(x_1=0,\ y_1=0)$에 놓고 다음 영상 I_2를 X축 선상이면서 두 영상간의 상위도 $D_{1,2}$만큼 떨어진 위치($(x_2, y_2=0)$에 놓는다. 이를 기준으로 두 영상과 I_3간의 상위도 $D_{1,3}$과 $D_{2,3}$을 만족하는 위치 (x_3, y_3)에 I_3를 배치한다. 3자의 영상일 때는 유일한 위치를 결정할 수 있지만, 4장 이상일 때는 상호간의 제약을 만족하는 위를 구할 수 없다. [그림 1-38]은 영상들 간의 분포도를 나타낸다. 유사한 영상은 서로 근거리에 위치하고 동일 형상을 지니면서 회전각도가 다른 영상은 동일 원주상에 존재한다.

[그림 1-37] 상위도 공간에서 4장의 영상을 배치시킨 모습



Collection @ chosun



[그림 1-38] 상위도에 따른 영상들의 분포

상위도 패턴을 이용한 인식처리의 과정으로 1)상위도 공간을 만드는 과정, 2)모델영상에 대한 상위도 패턴을 만드는 과정, 3)입력영상에 대한 상위도 패턴을 만드는 과정, 4)모델영상과 입력영상의 상위도 패턴의 비교에 의한 인식과정으로 크게 나누어진다. 1)과 2)의 과정은 시스템의 작동에 앞서 미리 계산해 놓고, 3)과 4)의 과정은 전용의 하드웨어를 이용하면 신속히 계산해 낼 수 있다. 인식은 4)의 과정에서 손의 형상을 정확히 표현하는 모델영상의 상위도 패턴과 입력영상의 상위도 패턴 사이의 차이를 계산하여 최소값을 나타내는 모델을 입력영상의 손 형상으로 인식한다.

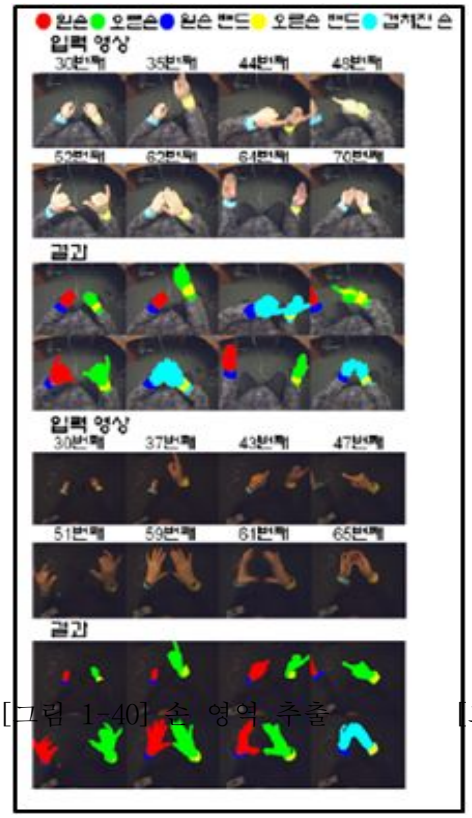
마. 고속카메라와 가속도 센서를 이용한 방법

고속카메라와 가속도 센서를 이용한 방법은 모자에 부착된 카메라와 손목에 착용한 가속 센서를 통해 사용자의 수화 동작을 관찰한다. 입력영상은 모자에 부착된 카메라를 통해 들어오고, 손의 움직임은 손목에 착용한 가속 센서를 통해 움직임 정보를 받는다. [그림 1-39](a)는 모자에 부착된 카메라로부터 획득한 영상이고, (b)는 손목에 착용될 가속 센서의 보드를 보여준다. [그림 1-40]과 [그림 1-41]은 추출된 손 영역을 보여주고 있다.

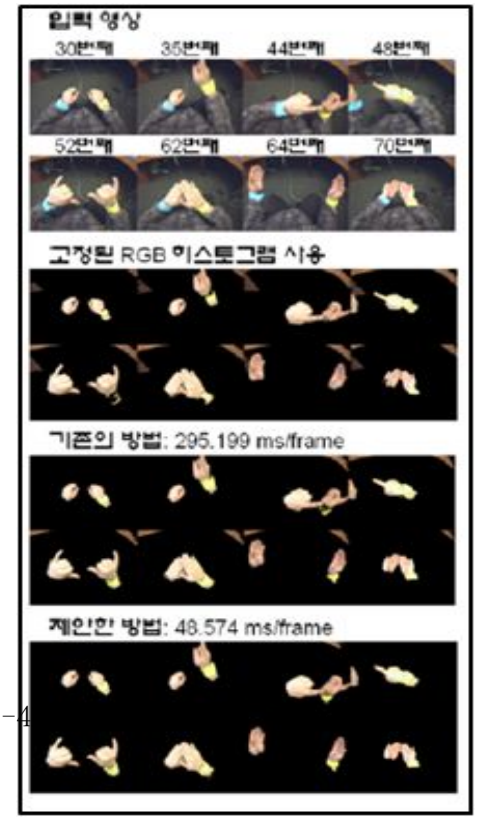
(a) 카메라로부터 획득한 영상 (b) 가속도 센서 보드

[그림 1-39] 카메라로 획득한 영상과 가속도 센서 보드

Collection @ chosun



[그림 1-40] 손 영역 추출



[그림 1-4

3. 연구의 목적 및 논문의 구성

최근 사용자 친화형 유저 인터페이스 구현 환경이 대두되면서 인간과 컴퓨터간의 원활한 상호작용의 필요성이 커짐에 따라 인간의 행동을 분석하는 연구가 많이 수행되고 있다. 인간은 정보전달을 위하여 언어 이외에 동작, 표정과 같은 비언어적인 수단을 이용한다. 이러한 비언어적인 수단을 정확히 분석할 수 있다면 인간과 컴퓨터간의 자연스럽고 지적인 인터페이스를 구축할 수 있게 된다. 인간은 다양한 수단을 사용하여 상대방에게 자신의 의사를 전달한다. 이 경우 흔히 언어적 수단만을 고려하기 쉬우나 이 외에도 몸짓, 손짓, 얼굴의 표정 등과 같은 비언어적 수단이 보다 많은 정보를 전달한다는 연구 결과가 보고되어 있다[2]. 인간과 의사소통이 자유스럽고 보다 지적인 시스템을 제작하기 위해서는 무엇보다도 이러한 비언어적 대화 수단을 자동화함이 필요하다. 그 중에서도 손은 인간에게 있어서 가장

Collection @ chosun

중요한 도구의 하나로 그 동작과 모양이 의미하는 바를 인식하는 것은 정보 시스템과 인간 사이의 자연스러운 인터페이스를 실현할 수 있다는 점에서 오랜 기간 동안 많은 사람들의 연구테마로 다루어져 왔다. 그러나 복잡한 환경으로부터 사람의 손 형상만을 정확히 추출하고 인식하여 의미 있는 입력정보로 사용하는 것은 어려운 일이다. 기본적으로 정확한 손 영역 분할을 위한 알고리즘이 필수이며 손 형상의 분석과 인식을 위한 빠른 처리 속도 또한 고려해야 할 요소이다.

본 논문에서 제안하는 방법은 이러한 문제점들을 해결하기 위하여 별도의 센서를 부착하지 않은 단일 카메라 환경에서 손 형상을 입력정보로 사용하여 템플리트 매칭에 의한 손가락 끝 점(Fingertip Point)을 인식하여 가상환경을 제어한다. 또한 인식한 손 형상을 신경망 알고리즘을 이용하여 수화인식을 위한 보다 안정적이며 강인한 손 형상 인식 시스템을 구현하고자 한다. 이는 비장애인과 장애인간의 상호 보다 편리하고 자연스러운 인터페이스를 제공한다. 비장애인이 장애인과 대화를 시도하려면 먼저 수화로 말하는 방법을 배워야한다는 단점이 있다. 이는 경제적인 면뿐 아니라 시간적인 면에서도 비효율적이라고 할 수 있다. 본 논문에서는 이러한 비효율적인 방법을 버리고 비장애인과 장애인간 상호 편리하게 대화할 수 있는 인터페이스를 제공하려 하며, 제안하는 방법으로는 카메라로 들어오는 영상을 입력영상으로 하여 피부색 정보를 이용해 배경으로부터 손 영역만을 추출한 후 추출된 손 영역의 형상을 인식한다(전처리과정으로 모델이미지의 사이즈와 압축 및 컬러에 대한 정보를 정규화시켰다). 인식된 손 형상 정보를 템플리트 매칭을 이용하여 손가락 끝 점(Fingertip Point)만으로 가상환경 제어하며, 또한 인식된 손 형상 정보에 신경망 알고리즘을 적용함으로써 손 형상 인식률에 대한 안전성과 정확성을 향상시켰다.

본 논문의 구성은 다음과 같다.

제 Ⅰ장에서는 손 형상에 대한 손 영역 검출과 손 형상 인식에 관한 연구 동향 및 논문의 목적과 관련 기술에 대해 기술하였다.

제 Ⅱ장에서는 신경망에 대한 이해와 신경망의 생물학적 구조 그리고 신경망 알고리즘에 대한 기술 동향에 대해 기술하였다.

제 Ⅲ장에서는 본 논문에서 제안하는 피부색 정보를 이용하여 손 영역을 검출하는 방법에 관해서 기술하였다.

Collection @ chosun

제 Ⅳ장에서는 본 논문에서 제안하는 신경망을 이용하여 손 형상을 인식하는 방법에 대해 기술하였다. 첫 번째, 실험을 위해 모델영상에 대한 전처리 과정이며 두 번째, 템플리트 매칭을 이용하여 손가락 끝 점(Fingertip Point)을 인식하고 가상환경을 제어하는 방법 세 번째, 신경을 이용하여 손 형상을 인식하는 방법에 대해 기술하였다.

제 Ⅴ장에서는 본 논문에서 제안한 템플리트 매칭을 이용한 가상환경 제어와 신경망을 이용한 손 형상 인식 시스템에 대한 실험 분석 및 평가 결과에 대해서 기술하였다.

제 Ⅵ장에서는 연구에 대한 결론 및 향후 추가적인 연구 과제를 제시하였다.

dCollection @ chosun

II. 신경망 알고리즘

1. 신경망의 개요

기존의 기호 처리 기반의 인공지능 분야와는 달리, 신경망은 인간의 뇌 기능을 모델링하여 구현한 정보처리 모델이다. 기존 디지털 컴퓨터는 결정론적인 2진 계산 모델에 기반하여 순차적으로 정보를 처리하는 방식이지만, 신경망은 생물학적인 방식의 모델로서 문제를 잘게 나누고 그것들의 결과를 더함으로써 문제를 해결하는 병렬적이고 아날로그적인 처리 방식이다.

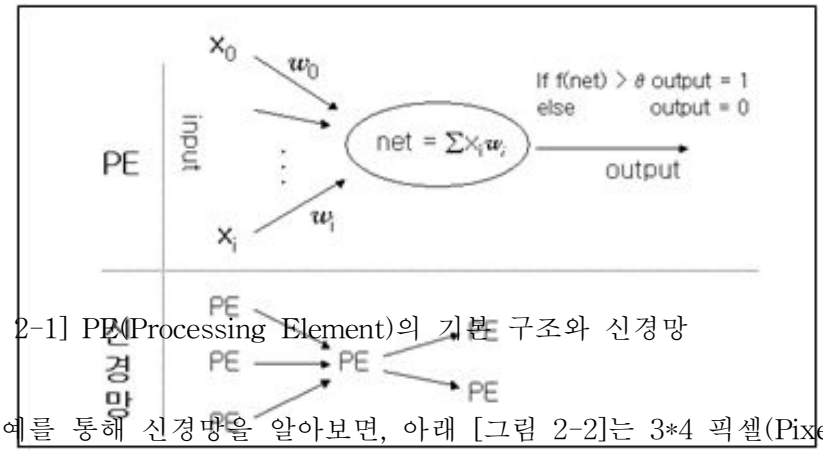
신경 조직의 기본구성은 뉴런(Neuron)이라 불리는 신경세포로 형성되어 있다. 인간의 두뇌는 약 10^{11}개의 뉴런으로 이루어져 있고 각 뉴런은 1,000에서 20,000개의 다른 뉴런과 서로 연결되어 있다. 수상돌기(Dendrite)는 다른 신경세포로부터 전기화학적 신호의 형태로 정보를 전송받아 신경세포체(Cell Body)에 전달하는 역할을 하며 신경세포는 이를 해석하여 축색돌기(Axon)를 통해 다른 신경세포에 정보를 전달하게 된다. 그리고 뉴런과 뉴런간의 연결접속부인 시냅스(Synapse)는 뉴런간의 연결강도(Connection Weight)를 나타낸다. 연결강도는 한 뉴런이 다른 뉴런을 흥분 또는 억제시키는 정도를 나타낸 것으로 이 연결강도에 의해서 지식을 표현하고 학습이 이루어진다(신경망에서 학습이란 원하는 결과의 노드가 최대값이 나오도록 적절한 가중치를 설정하는 과정이다).

신경망 모델은 1946년 워렌 맥클로츠(Warren McCulloch)와 월터 피츠(Walter Pitts)의 논문에서 처음 제안되었으며 기존의 기호처리 인공지능 모델과 같은 시기에 연구가 시작되었다. 초기에 신경망은 인공지능 연구자들로부터 많은 주목을 받았지만 신경망의 한 모델인 “퍼셉트론”의 한계가 지적되면서 인공지능 연구의 주도권이 기호처리 방식으로 넘어가게 되었다. 하지만 그로부터 몇 십 년 후에 “퍼셉트론”의 한계를 극복한 “다층 퍼셉트론”이 발표됨으로써 신경망은 기존의 기호처리 방식과 견줄 만한, 또는 그 한계를 극복할 수 있는 새로운 대안으로 각광을 받고 있다.

신경망은 인간의 뇌 구조를 이용하여 모델링된 알고리즘으로 패턴인식 등에 그

Collection @ chosun

활용 범위가 대단히 광범위하며 특히 영상처리 분야에서는 빼놓을 수가 없다. 또한 물질의 성질을 잃지 않는 최소 단위가 분자인 것처럼 인간의 뇌를 포함한 신경들을 구성하는 최소 단위는 뉴런이다. 뉴런과 뉴런의 연결은 시냅스(Synapse)라는 물질이 관여한다. 만약 인간이 “통증”을 느꼈다면 그 통증은 뉴런들의 연결을 통해 뇌까지 전달되고, 뇌에서 그 통증을 느끼는 것이다. [그림 2-1]은 신경망의 최소 단위가 되는 PE(Processing Element)이다. 뉴런을 모델링했다고 할 수 있다. 입력의 각 성분에 가중치(Weight)를 곱한 값인 net 값이 되면 출력은 net 값과 함수 f에 따라 달라진다.



[그림 2-1] PE(Processing Element)의 기본 구조와 신경망

숫자를 인식하는 예를 통해 신경망을 알아보면, 아래 [그림 2-2]는 3*4 픽셀(Pixel)로 구성된 숫자 7과 신경망 구성이다. 숫자 7을 인식하기 위해서 if - then으로 구성할 수도 있고, Table 형식으로 구성할 수도 있다. if - then 으로 구성한다면

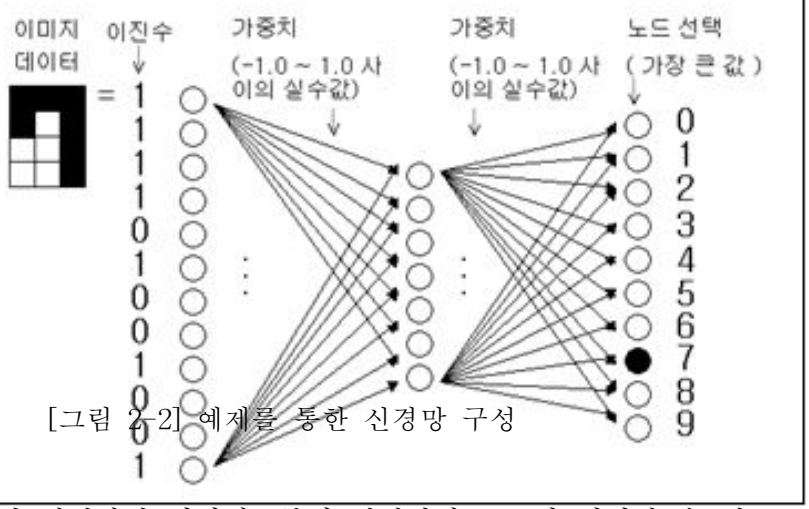
```
x = GetBinaryFromImage();
```

과 같이 될 수 있을 것이다. 그러나 노이즈가 발생하여 노이즈로 인해 2진수가 111001001001로 바뀌었다고 하면 모든 경우의 수를 if -then으로 표현하기란 쉽지

```
if( x == 111101001001 )
    then return 7;
```

Collection @ chosun

않다. 하지만 신경망은 하나의 알고리즘으로 분류(Classification)가 가능하다. 신경망이 노이즈에 강하다는 말은 그런 이유 때문이다.



[그림 2-2] 예제를 통한 신경망 구성

[그림 2-2]를 좀 더 설명하면 이미지로부터 신경망의 노드가 처리할 수 있는 데이터로 변환한 다음 각 입력 노드 값으로 설정하고 가중치(Weight)가 적절히 설정되었다면 출력 노드에서 7이라는 노드가 가장 큰 값으로 표현될 것이다. 임의의 이미지를 입력으로 하였을 때 가장 큰 값의 출력 노드가 그 이미지가 가리키고 있는 숫자인 것이다. 초기 신경망의 가중치 값은 보통 -1.0 ~ 1.0의 임의의 값으로 설정되어 있다.

신경망에서의 학습이란 원하는 결과의 노드가 최대값이 나오도록 적절히 가중치를 설정하는 과정이다. 쉽게 말하면 111101001001이나 111001001001의 입력이 들어가도 출력 노드 7이 최대값이 되도록 가중치를 설정하면 된다. 학습에 사용되는 대표적인 알고리즘이 바로 역전파(Back Propagation) 알고리즘이다.

2. 신경망의 생물학적 구조

인간의 두뇌는 평균 천만개, 즉 개의 신경세포들로 이루어져 있으며, 각 신경세포는 수상돌기들을 통해서 대략 10,000개까지의 다른 신경세포들과 연결되어 있

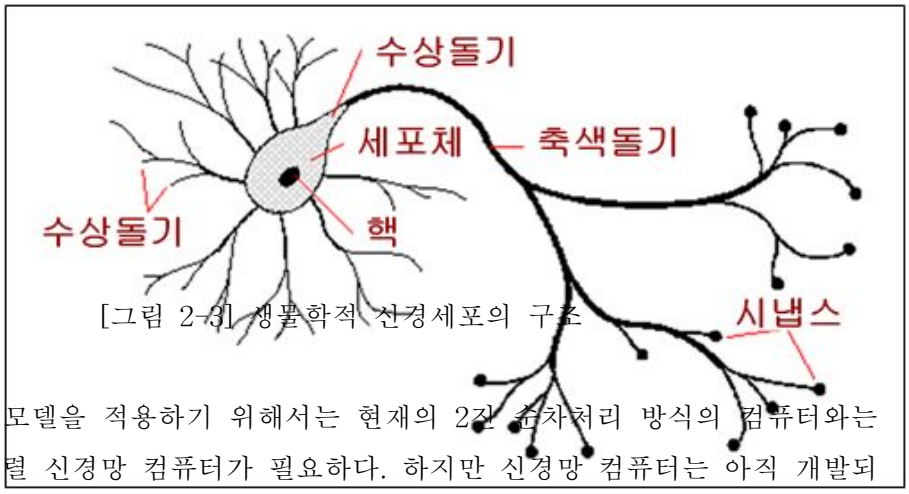
10^{11}

Collection @ chosun

다. 신호는 전기화학적인 과정들을 통해서 전달되는데, 수상돌기와 축색돌기의 연결부인 시냅스가 기본적인 신호전달 지점이다.

뉴런은 신경조직의 기본단위로서 수상돌기, 신경세포체, 축색돌기로 구성되어 있다. 수상돌기는 다른 뉴런들의 축색돌기로부터 전기화학적 신호를 받아 신경세포체에 조합의 형태로 전달하는 역할을 하고, 신경세포체는 뉴런 조절기관으로 수상돌기로부터 받은 신호를 해석하여 축색돌기를 통해 정보를 전달한다.
하나의 뉴런은 다른 뉴런과 연결되어 있어 정보 채널을 통해 계속적으로 정보를 접수하여 해석하고 전송한다.

인간 두뇌의 지능은 이처럼 다량으로 연결된 구조이기 때문에 가능하며, 뉴런간의 연결강도, 연결형태에 의한 것이다. 학습은 뉴런간의 연결강도 변화에 의해 일어난다. 연결강도는 증가될 수도 있고 감소될 수도 있는데 이러한 기능은 수상돌기와 축색돌기 사이의 공간인 시냅스에서 일어난다. 이러한 학습에 따라 변할 수 있으므로 여러 가지 정보들을 계속적으로 습득할 수 있는 것이다. [그림 2-3]은 인간 두뇌의 기본 단위인 신경세포의 생물학적 구조이다.



[그림 2-3] 생물학적 신경세포의 구조

실제 신경망 모델을 적용하기 위해서는 현재의 2진 순차처리 방식의 컴퓨터와는 다른 구조의 병렬 신경망 컴퓨터가 필요하다. 하지만 신경망 컴퓨터는 아직 개발되지 않았으며 현재는 기존의 컴퓨터에서 병렬구조의 신경망을 소프트웨어적으로 순차적 구조의 시뮬레이션으로 구현하고 있다. 그렇기 때문에 병렬처리 기반의 빠른

Collection @ chosun

속도가 불가능하여 처리속도가 현저히 저하되고 있다. [표 2-1]과 [표 2-2]는 뉴런의 구조 및 생물학적 신경망의 특성을 간략화한 것이다.

[표 2-1] 뉴런의 구조

뉴런의 구조

세포체(Soma)

-신경세포의 중앙에 위치

-정보의 수용, 연산처리, 출력의 전송

수상돌기(Dendrite)

-나무가지처럼 넓게 퍼져 있는 두껍고 짧은 다수의 돌기

-다른 뉴런과 연결되어 입력신호 받음

축색돌기(Axon): 한 개의 가느다란 섬유

-세포체에 붙어 있으며 전기적으로 활성화되고, 뉴런에 의해 발생되는 펄스를 다른 뉴런들에 전달

-축색의 끝부분은 가느다란 가지로 나누어져 있고, 다른 뉴런의 수상돌기와 시냅스(Synapse)를 통해 연결

[표 2-2] 생물학적 신경망의 특성

생물학적 신경망의 특성

병렬적 계산과정, 분산저장

연상기억(Associative Memory) 능력

-일정한 범용패턴을 연결강도로 저장하였다가 미지의 입력패턴이 주어질 때 가장 유사한 패턴 찾아냄. 영상이나 신호패턴들의 일부 또는 단서들로부터 전체적인 표현을 추출

Collection @ chosun

학습능력(자기조직화, Self Organizing)
-주어진 입력에 대해 자신의 내부 구조를 스스로 조직해 나감으로써 학습해 나가는 능력
-입력 패턴에 대해 정확한 해답이 없어도 스스로 학습할 수 있는 능력

3. 뉴런의 모델링

신경회로망 모델의 기본 단위는 하나의 처리단위(Processing Element)로서 입력(Input), 연결강도(Weight), 연결함수(Combining Function), 전이함수(Transfer Function), 출력(Element Output)이라는 5가지의 주요 부분으로 구성되어 있다. 인간의 신경 시스템과 신경회로망 모델을 비교하면 [표 2-3]과 같다.

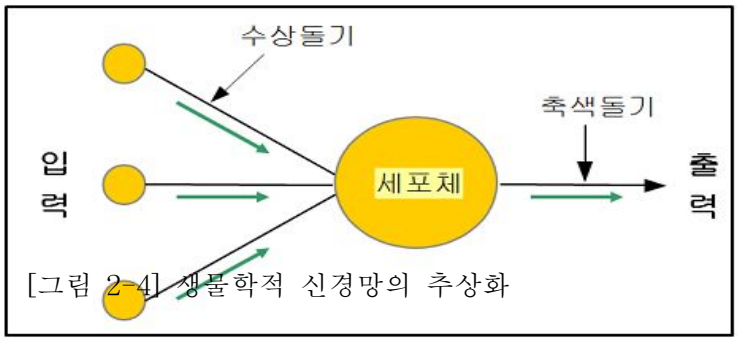
[표 2-3] 신경 시스템과 신경회로망 시스템의 비교

신경 시스템	신경회로망 시스템
뉴런(Neuron)	처리단위(Processing Element)
수상돌기(Dendrite)	연결함수(Combining Function)
신경 세포체(Cell Body)	전이함수(Transfer Function)
축색 돌기(Axon)	출력(Element Output)
시냅스(Synapse)	연결 강도(Weight)

입력을 통하여 정보가 처리단위에 들어오는데 정보의 Source는 다른 처리 단위일 수도 있고 외부일 수도 있다. 연결 강도는 어떤 정보의 입력이 그 처리 단위에 얼마나 영향을 미칠 것인가를 결정한다. 한 처리단위로부터 다른 처리단위로 보내지는 신호의 강도는 연결강도에 의해 수정된다. 연결함수는 입력과 정보와 연결강도를 조합한다. 가장 일반적인 연결함수는 Weight Sum의 형태를 취하는 것이다. 전이함수는 연결함수의 결과를 해석하여 그 처리단위의 출력값을 결정한다. 전이함수에는 Mard Limit, Threshold Logic, Sigmoid 등이 있다. 처리단위의 출력은 전이함수의 결과이며, 하나의 처리단위에서 오직 하나의 출력 값만이 존재한다. 하나 이상의 처리단위들이 모여 층(Layer)을 이루는데 입력층, 출력층, 은닉층이 있다. Interactive한 구조 덕분에 인간의 두뇌와 마찬가지로 신경회로망의 지능과 정보의

Collection @ chosun

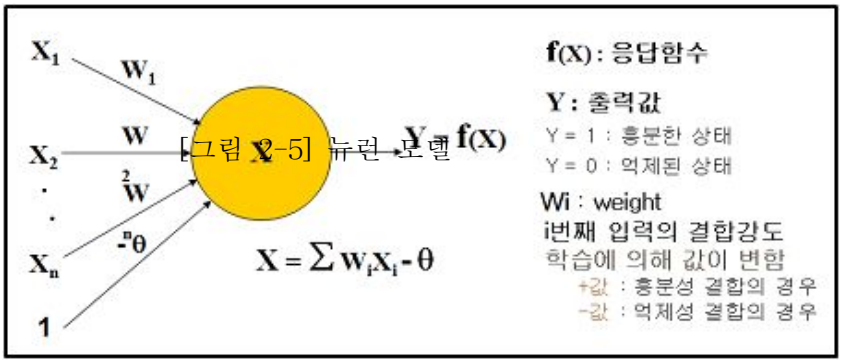
저장능력을 병렬구조로 형성된 정보가 분산되어 표현된다. [그림 2-4]는 생물학적 신경세포의 구조를 추상화한 것이다.



[그림 2-4] 생물학적 신경망의 추상화

4. 신경망 모델의 구조 및 장·단점

간단한 하나의 신경망의 구조는 아래 그림과 같다. n개의 X 입력들이 각각 W라는 가중치에 의해 X라는 뉴로드(인공 신경세포)에 연결되어 있다. 각 입력들은 각각의 가중치들과 곱해져 더해지고, 이 결과가 특정한 조건보다 크다면 출력을 통해 전달된다. 여기서 가장 중요한 점은 가중치 W들을 어떻게 설정하는가 이다. 특정한 입력이 들어왔을 때 특정한 출력이 나오도록 하기 위해, 그 조건에 맞도록 적절하게 가중치들을 조정하여야 한다. 이처럼 특정 상황에 맞도록 가중치를 설정하는 것이 바로 정보의 저장, 또는 학습이라 할 수 있다. 보통 여러 개의 입력들을 반복적으로 신경망에 넣음으로써 점차적으로 가중치들을 원하는 방향으로 조절하게 된다. [그림 2-5]는 추상화한 신경세포의 구조를 인공신경망인 뉴런 모델로 변경한 것이다.



[그림 2-5] 뉴런 모델

Collection @ chosun

가. 기호처리 방식과 비교한 신경망의 장점

첫째, 신경망은 사람의 뇌에 더욱 근접한 구조를 가지고 있다. 일반적으로 비슷한 구조에서 비슷한 기능이 생겨날 거라고 기대할 수 있다. 그러므로 신경망의 특징인 학습능력, 대규모 병렬동작, 지식의 분산표현 등은 지금까지 인공지능의 구현에 있어 문제가 되었던 모호함의 처리, 탐색공간의 거대화 등을 해결할 수 있는 가능성을 제시하고 있다.

둘째, 신경망은 학습을 위한 적당한 구조를 가지고 있다. 기존의 방식으로도 학습하는 시스템을 만들 수 있지만, 이를 위해 많은 노력이 소요된다. 그러나 신경망의 경우, 사람은 단지 신경망으로 하여금 가르치기를 원하는 것들을 경험하도록 하기만 하면 되므로 학습이 훨씬 용이하다.

셋째, 신경망은 매우 단순하다. 신경망을 구성하는 신경세포들의 구조와 기능은 매우 단순하여 복잡한 프로그래밍 과정이 필요가 없다. 그러나 이처럼 단순한 구조로부터 여러 가지 복잡한 작업이 가능하다.

나. 기호처리 방식과 비교한 신경망의 단점

첫째, 신경망의 내적 지식을 사람이 이해하기 힘들다는 점이다. 기존의 방식은 사람에 의해 프로그래밍 됨으로 이를 분석하면 어떻게 동작할 것인지를 알 수 있지만, 신경망은 학습 후 회상을 통해서만 결과를 알 수가 있다.

둘째, 아직까지 신경망의 하드웨어적인 구현이 어렵다는 점이다. 신경망의 지식은 그것의 연결 가중치의 전체적인 상태에 의해 결정된다. 따라서 보다 많은 지식을 가지기 위해서는 보다 많은 연결이 필요하게 된다. 그러나 아직까지의 반도체 기술로는 충분히 많은 연결을 구현하기가 어렵다.

셋째, 신경망은 사람이 잘하는 여러 가지 작업에 있어 컴퓨터보다 뛰어난 성능을 발휘하지만, 사람의 뇌가 컴퓨터에 비해 잘하지 못하는 산술 연산과 같은 부분에 있어서는 컴퓨터에 미치지 못한다. 실제로 산술적인 연산 기능을 수행하는 신경망을 구성하기는 매우 힘들다.

Collection @ chosun

[표 2-4]는 신경망컴퓨터와 디지털컴퓨터에 대해 비교한 것이며, [표 2-5]는 신경망 시스템의 응용분야를 나타낸 것이다.

[표 2-4] 신경망과 디지털 컴퓨터의 비교

	신경망컴퓨터 (Connectionism)	디지털컴퓨터 (Symbolism)
처리데이터	디지털,아날로그	디지털
처리 방식	병렬처리	직렬(순차)처리
기본 소자	뉴런	논리소자
실행 근거	학습	프로그램
정보 저장	뉴런간의연결강도	기억장치
정보 검색	내용에의함	주소지정
응용 분야	연상,추론,인식	복잡한계산

[표 2-5] 신경망의 응용 분야

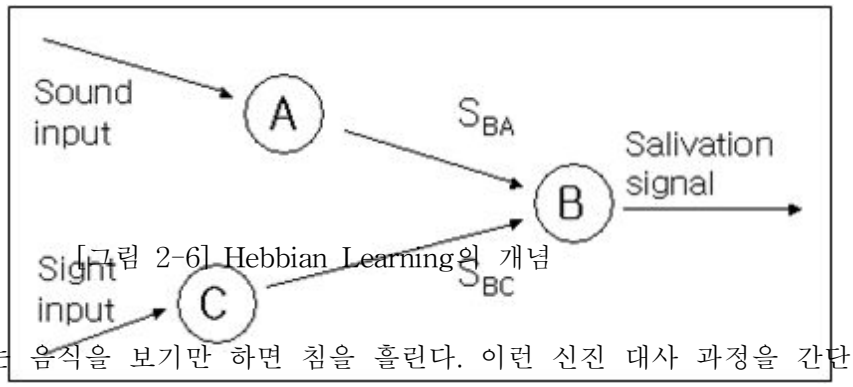
응용분야	-고도 병렬 분산처리를 요하는 복잡한 시스템에 관한 분야 -기존의 폰노이만 컴퓨터가 잘 하지 못하는 분야	
사람의 정보처리에 관련된 분야	화상처리	문자인식, 패턴인식, 화상압축
	음성처리	음성인식, 음성합성
	자연어처리	자연언어 해석, 기계번역, 키워드 추출
	운동제어	로봇제어
사회시스템, 산업시스템에 관련된 분야	전문가시스템	진단시스템, 수요예측시스템
	최적화	경로선택
	제어	Process제어, 적응제어시스템, 공장제어

Collection @ chosun

5. 신경망 알고리즘에 대한 관련 연구

5-1. Hebb의 학습

신경망은 한 사람에 의해 그 모든 알고리즘이 정립된 것이 아니다. 오랜 시간동안 여러 사람에 의해 수정되고 보완된 것이다. 최초 신경망의 기본 개념을 제시한 학습론이 바로 Hebbian Learning이라 볼 수 있다. Hebb는 1949년에 "Organization of Behavior"라는 출간물에서 "셀 A가 셀 B를 자극시키는 데 충분하고, 지속적인 원인으로 작용한다면, 셀 B에 대한 셀 A의 작용력이 내부적으로 변화를 일으켜 증가하게 된다." 라고 했다. 이해를 돕기 위해 그림으로 요약하면 [그림 2-6]과 같다.



[그림 2-6] Hebbian Learning의 개념

예를 들면, 개는 음식을 보기만 하면 침을 흘린다. 이런 신진 대사 과정을 간단히 Hebb는 셀 C (또는 셀들)이 셀 B를 자극한 것으로 도식화한 것이다. 그리고 음식을 줄 때, 종을 울리고 그러한 과정이 지속된다면, 후에는 그 개는 종소리만 들어도 침을 흘리게 된다. 결국에는 음식을 주지 않고 종만 울려도 침을 흘리게 되는데 셀 A가 셀 B에 작용하기 때문이다. 처음에 개의 뇌 구조는 종소리와 침을 흘리는 반응과는 전혀 관계가 없었으나, 그렇게 되도록 뇌 구조 또는 뇌를 구성하는 뉴런들의 연결 구조가 바뀐 것이라 할 수 있다. 이런 학습을 Hebbian Learning이라고 한다.

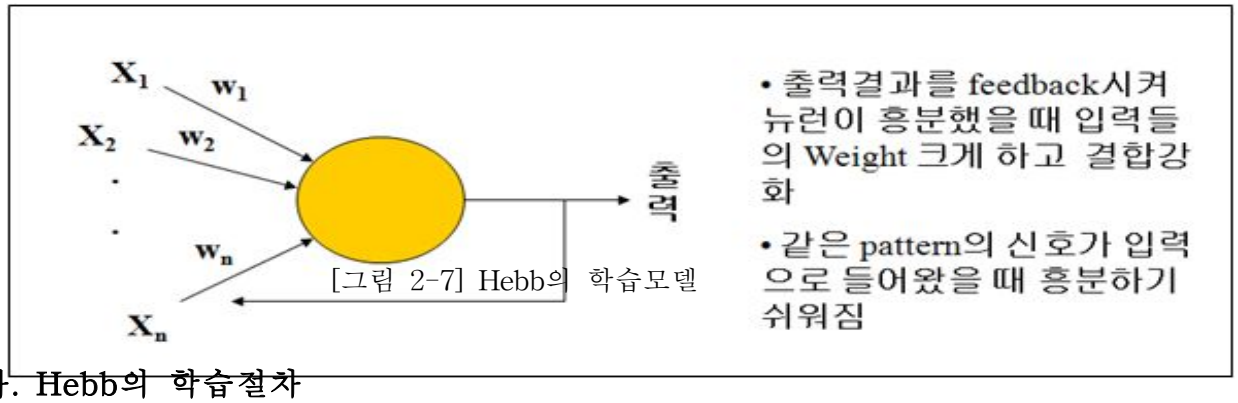
최초에 그런 행동 패턴을 몇 개의 셀과 그 연결로 간단히 도식화를 시도했다는 점에서 높이 가치를 부여할 만한 것이 되었고, 이러한 개념이 신경망의 근본을 이

Collection @ chosun

루는 개념이 되었던 것이다. Hebbian Learning은 알고리즘 측면에서보다 최초 개념 제시라는 측면에서 그 중요성이 있다.

가. Hebb의 학습규칙

심리학자 Hebb(1949)는 뇌의 학습모델에 대해 제안을 하였으며, “Hebb의 시냅스”라고 알려진 시냅스의 연결강도 조정을 위한 생리학적 학습규칙을 기술하였다. Hebb의 학습모델은 [그림 2-7]과 같다.

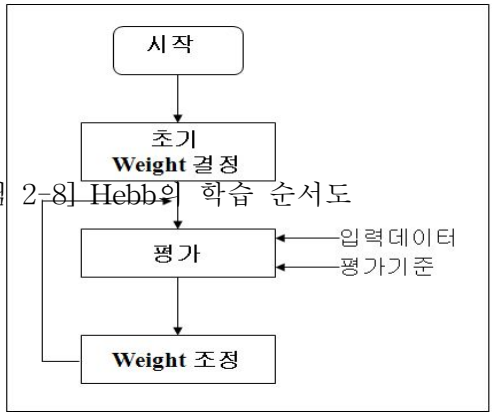


[그림 2-7] Hebb의 학습모델

나. Hebb의 학습절차

Hebb의 학습절차는 아래와 같은 절차를 갖으며, 도식화 하면 [그림 2-8]과 같다.

1) 초기치에 대한 지식이 없으면 랜덤하게 결정
2) 학습 데이터를 입력하여 주어진 평가기준을 적용하여 평가
3) 평가결과에 따라 Weight 조정
4) 반복 적용, 최적치로 접근

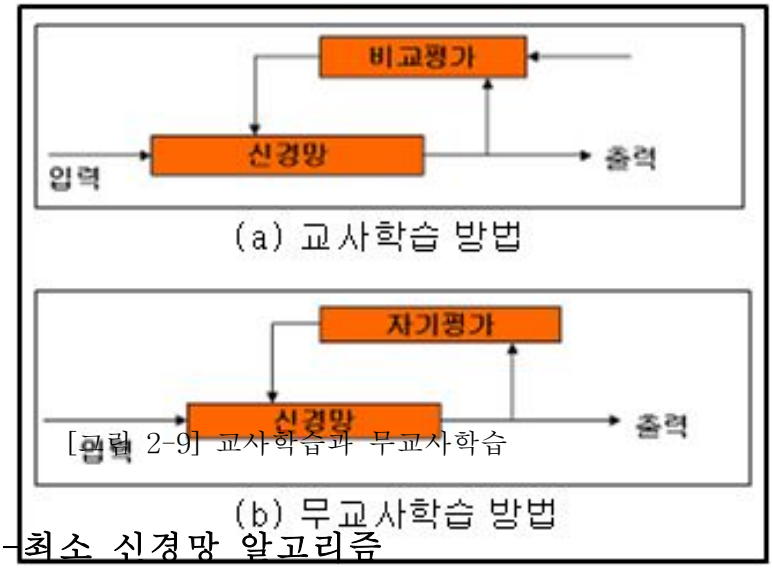


[그림 2-8] Hebb의 학습 순서도

Collection @ chosun

다. Hebb의 학습방법

Hebb의 학습방법으로는 교사학습(Supervised Learning)과 무교사학습(Unsupervised Learning)이 있다. 교사학습 방법은 외부에서 교사신호(Teaching Input)로써 입력신호에 대한 정답 출력을 주는 학습방법이며, 무교사학습 방법은 평가기준은 있으나 일일이 교사 신호를 주지 않는 학습 방법이다. [그림 2-9]는 교사학습과 무교사학습 방법을 나타낸다.



[그림 2-9] 교사학습과 무교사학습

5-2. 퍼지 최대-최소 신경망 알고리즘

퍼지 최대·최소 신경망(Fuzzy Min-Max Neural Network, 이하 FMMNN이라고 함)은 1992년 Simpson에 의해 제안된 패턴 분류 방법이다. 이 방법에서는 입력 패턴 값들을 n차원의 패턴 공간 내에서 정의되는 초상자(Hyper-box)를 사용하여 나타내고 이 초상자들은 최소값, 최대값 그리고 이 최소값과 최대값으로 표현되는 멤버쉽 함수로 구성된다. 학습과정에서 입력 패턴에 대응하는 초상자들의 확장, 중복, 축소과정을 거쳐 최종 초상자들의 집합을 얻을 수 있다. 그리고 테스트 입력 패턴이 인가되면, 각 초상자들의 멤버쉽 함수를 이용하여 인가된 테스트 패턴과 초상자의 일치정도를 멤버쉽 값으로 구할 수 있으며, 가장 높은 멤버쉽 값을 갖는 초상자를 찾아 그 초상자의 클래스를 선정함으로써 패턴분류를 할 수 있게 된다. [표 2-6]은 FMMNN의 학습과정을 보여준다.

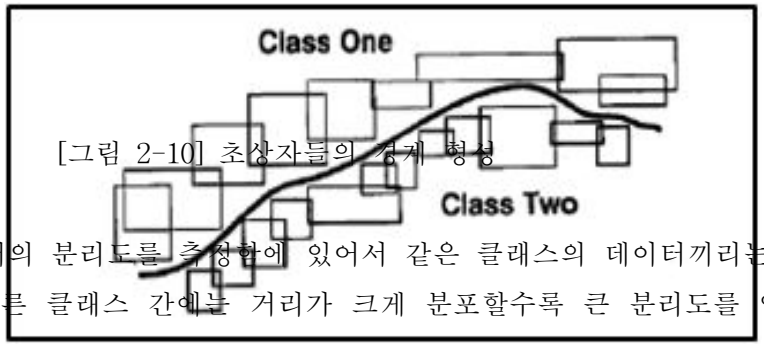
Collection @ chosun

[표 2-6] FMMNN의 학습과정

1. 확장(Expansion) - 입력 패턴이 인가되었을 경우 같은 클래스를 가지는 초상자들 중에서 확장기준(Expand Criterion)을 만족하는 초장자와 확장, 만약 확장할 초상자가 없다면 새로운 초상자 생성
2. 중복(Overlap) - 확장된 초상자가 다른 클래스의 초상자들과 겹침이 있는지를 검사
3. 축소(Contraction) - 중복 검사에서 서로 다른 클래스에 속하는 초상자들과 겹침이 있는지를 검사

퍼지 최대·최소 신경망은 비선형 분리 및 클래스의 중복이 가능하여 비선형 패턴 분류가 가능[31]하며, 비모수적 방법이기 때문에 입력데이터의 확률분포를 모르는 경우에도 확률분포에 대한 가정없이 적용할 수 있다[31]. 대부분의 비선형 패턴 분류기들이 학습을 위해 많은 시간이 소요되지만, FMMNN의 경우 빠르고 효율적인 특성을 보이기 때문에[31] 실시간 패턴 분류 시스템에 많이 사용되고 있다.

FMMNN은 비모수적인 특성을 가지고 있고 중복성을 인정하기 때문에 입력 데이터 패턴을 초상자들의 집합으로 나타내더라도 초상자들의 경계를 이용해 주어진 데이터의 분포를 반영할 수 있다[그림 2-10]. 또한 초상자들은 여러 입력 데이터들을 내포할 수 있기 때문에 데이터의 정보를 최대한 유지하면서 작은 수의 초상자들로 데이터를 표현할 수 있는 장점이 있다.

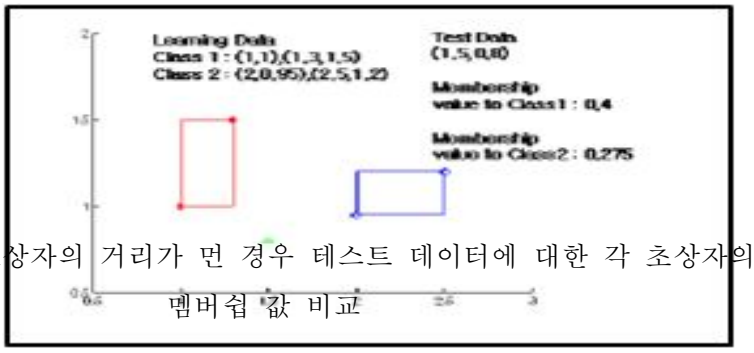


[그림 2-10] 초상자들의 경계 형성

일반적으로 데이터의 분리도를 측정함에 있어서 같은 클래스의 데이터끼리는 인접하게 분포하고 다른 클래스 간에는 거리가 크게 분포할수록 큰 분리도를 얻을

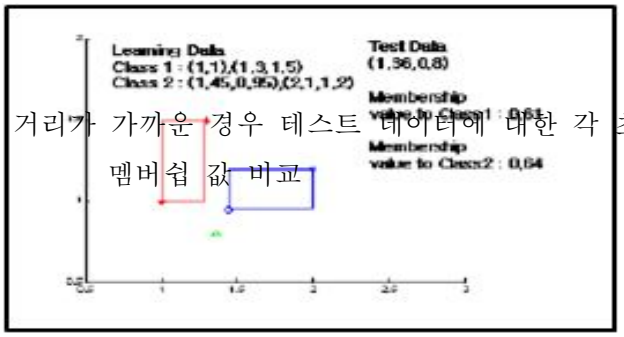
Collection @ chosun

수 있다. 이는 서로 다른 클래스의 데이터가 인접하게 분포할수록 데이터를 분류함에 오차가 커짐을 의미하며, 아래의 예를 통해 이를 확인할 수 있다. [그림 2-11]에서 볼 수 있듯이 2차원의 패턴 공간에서 두 초상자의 거리가 먼 경우, (1.5, 0.8)의 테스트 패턴이 인가되면 클래스 1 초상자는 0.4의 멤버쉽 값을 가지고 반면, 클래스2 초상자는 0.275의 멤버쉽 값을 가지게 된다. 이 경우 테스트 패턴은 멤버쉽 값이 큰 클래스 1을 선택하게 된다.



[그림 2-11] 두 초상자의 거리가 먼 경우 테스트 데이터에 대한 각 초상자의 멤버쉽 값 비교

하지만 [그림 2-11]에서처럼 테스트 패턴이 x축에 대해 클래스1의 초상자에 더 가깝게 인가되지만 두 초상자가 인접한 경우[그림 2-12] y축의 멤버쉽 값이 더 크게 작용을 하게 된다. 그래서 클래스1 초상자에 대한 테스트 패턴의 멤버쉽 값 ((0.82+1.22)/2 = 0.61)보다 클래스 2 초상자에 대한 멤버쉽 값((0.73+1.28)/2 = 0.64)이 더 크게 되어 테스트 패턴은 클래스 2로 분류가 된다.



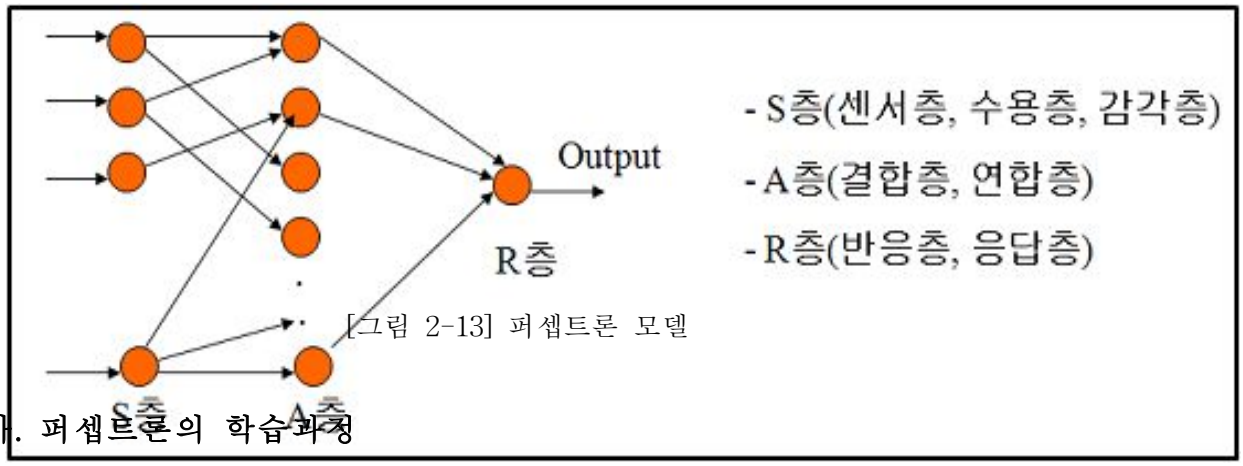
[그림 2-12] 두 초상자의 거리가 가까운 경우 테스트 데이터에 대한 각 초상자의 멤버쉽 값 비교

Collection @ chosun

이 방법을 적용하여 서로 다른 클래스에 속한 초상자들의 인접성을 구하여 인접성이 작은 특징들의 집합을 선택함으로써 데이터의 분류오차는 줄이고 데이터의 분리도는 높일 수 있는 방향으로 특징 집합을 선택하는 연구도 진행되었다[32].

5-3. 퍼셉트론(Perceptron)

퍼셉트론(Perceptron)은 심리학자인 Rosenblatt에 의해 1957년에 제안되었으며, 목적은 Pattern 분류를 위해 제안된 모델이다. 퍼셉트론은 학습에 의해 인식하는 능력을 향상시킬 수 있는 장치이지만 Pattern의 본질적인 특징에 대한 지식은 필요없다. 퍼셉트론모델은 수용층, 연합층, 반응층 3개의 층을 이루고 있으며 [그림 2-13]과 같다.



[그림 2-13] 퍼셉트론 모델

가. 퍼셉트론의 학습과정

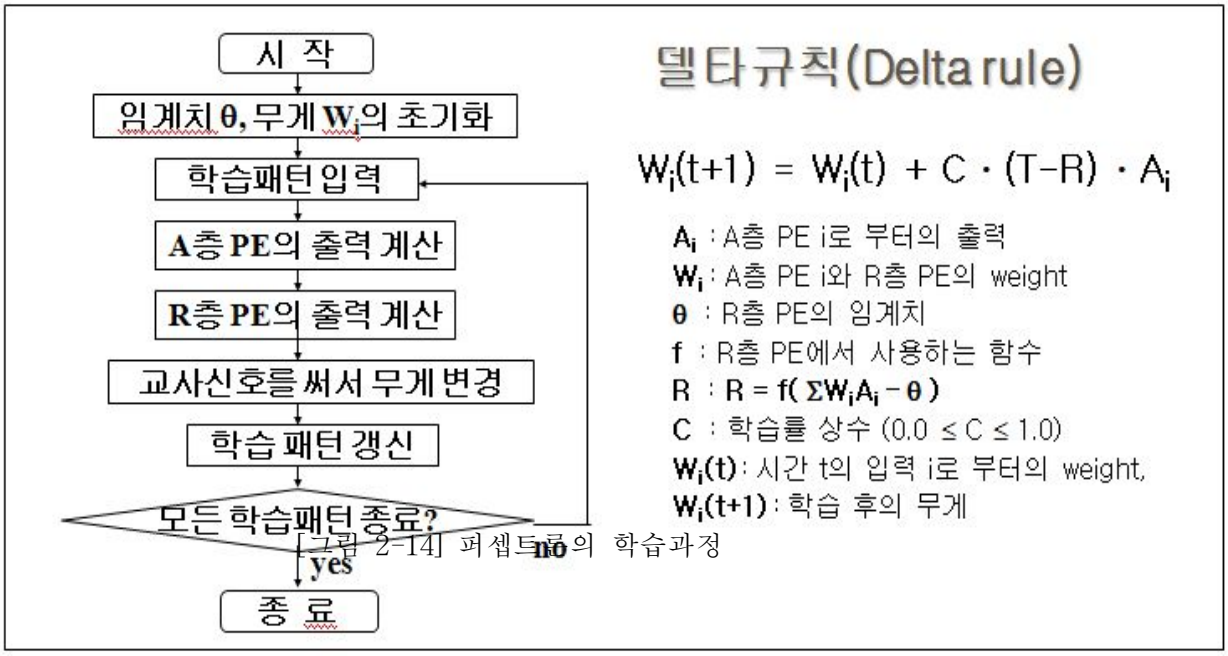
퍼셉트론의 학습과정은 R층 PE(Processing Element)의 출력 R과 교사신호 T를 써서 Weight를 바꾸는 과정이다. 퍼셉트론의 학습과정을 도식화하면 [그림 2-14]와 같다.

나. 퍼셉트론의 한계

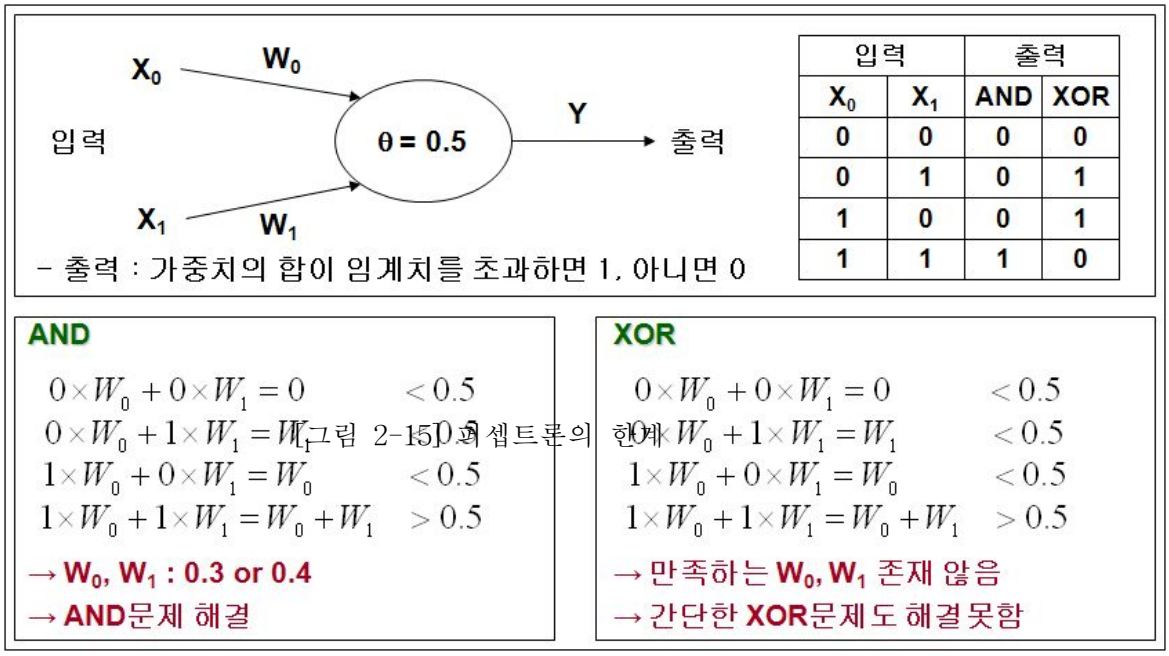
퍼셉트론의 한계는 입력값과 가중치의 값을 연산하여 출력값을 내는데 있어 "AND"연산의 문제는 해결하였지만, 간단한 "XOR"연산의 문제도 해결하지 못한다. 퍼셉트론이 “XOR”연산이 불가능한 것은 “XOR” 문제에서 선형 분리가 불가능하기

Collection @ chosun

때문이다. 이러한 문제를 해결하기 위해 20년이란 시간이 흘렀으며 그 해결책으로 나온 것이 바로 다층 퍼셉트론(Multi-Perceptron)이다. [그림 2-15]는 퍼셉트론의 한계를 보여주고 있다.



[그림 2-14] 퍼셉트론의 학습과정



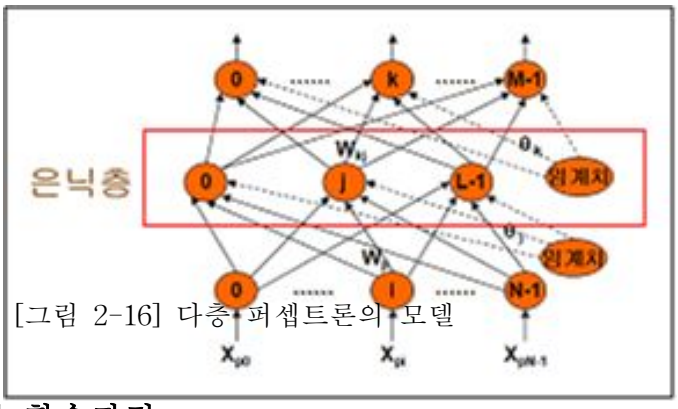
입력		출력	
X_0	X_1	AND	XOR
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

[그림 2-15] 퍼셉트론의 한계

Collection @ chosun

5-4. 다층 퍼셉트론(Multilayer Perceptron)

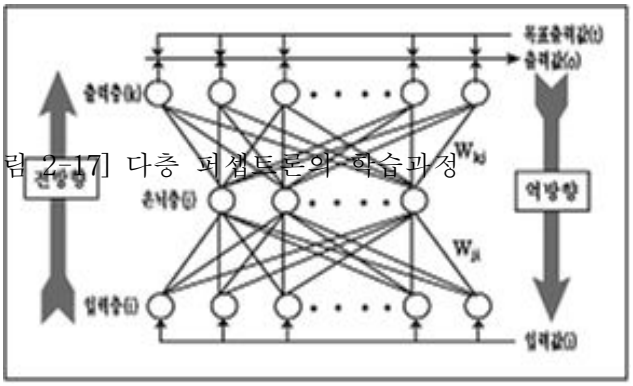
다층 퍼셉트론(Multilayer Perceptron)은 입력층과 출력층 사이에 하나 이상의 중간층(은닉층, Hidden Layer)이 존재하는 신경망으로서 기존의 단층 퍼셉트론의 문제점인 “XOR”문제를 해결하기 위한 방안으로 제안된 모델이다. [그림 2-16]은 다층 퍼셉트론의 모델이다.



[그림 2-16] 다층 퍼셉트론의 모델

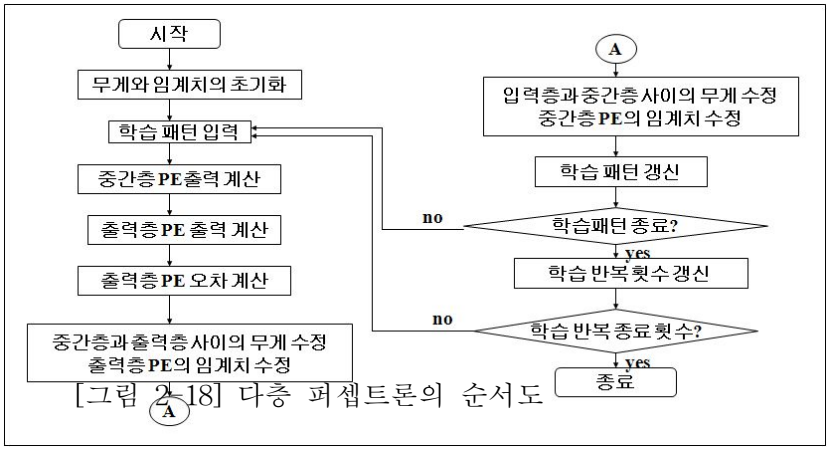
가. 다층 퍼셉트론의 학습과정

단층퍼셉트론의 문제점을 해결하기 위한 효과적인 학습알고리즘이 없어서 60, 70년대에는 사용되지 않았다. 80년대 중반 오류 역전파 알고리즘(Error Back Propagation Algorithm) 학습규칙과 일반화된 델타규칙(Generalized Delta Rule)에 의해 학습되며, 알고리즘은 원하는 목표값과 실제 출력값 사이의 오차제곱합으로 정의된 비용함수 e값을 경사 하강 추적법에 의해 최소화하는 방향으로 학습된다. 그림으로 표현하면 [그림 2-17]과 같고, 순서도는 [그림 2-18]과 같다. [표 2-7]은 다층 퍼셉트론의 학습절차이다.



[그림 2-17] 다층 퍼셉트론의 학습과정

Collection @ chosun



[그림 2-18] 다층 퍼셉트론의 순서도

[표 2-7] 다층 퍼셉트론의 학습 절차

1단계	가중치와 임계치를 초기화
2단계	입력과 목표 출력을 제시
3단계	제시된 입력 벡터를 이용하여 은닉층 j번째 뉴런으로의 입력값 계산
4단계	시그모이드 함수를 사용하여 은닉층의 출력을 계산
5단계	은닉층의 출력을 이용하여 출력층 뉴런 k로의 입력값을 계산
6단계	시그모이드 함수를 사용하여 출력층의 출력을 계산
7단계	입력패턴의 목표출력과 실제출력과의 오차값을 계산하고 출력층 오차합을 학습패턴의 오차로 누적 시킨다.
8단계	출력층 오차값과 은닉층과 출력층의 가중치값을 이용하여 은닉층의 오차를 계산한다.
9단계	4단계와 7단계에서 구한 은닉층 유런 j의 출력값과 출력층의 오차값을 사용하여 출력층의 가중치를 갱신한다(임계치도조정)
10단계	출력층에서와 마찬가지로 입력층과 은닉층의 가중치 값과 임계치 값을 갱신한다.
11단계	모든 학습패턴에 대하여 전부 학습할때까지 2단계로 분기하여 반복수행한다.
12단계	출력층의 오차값 e가 허용값 이하이거나 최대반복횟수보다 크면 종료 그렇지 않으면 2단계로 가서 반복한다.

또한 실제 응용 가능한 분야에 초점을 맞추어 시스템환경을 제한함으로써 일반적으로는 접근하기 어려운 위의 문제들을 해결하려는 연구도 진행되었다[35].

Collection @ chosun

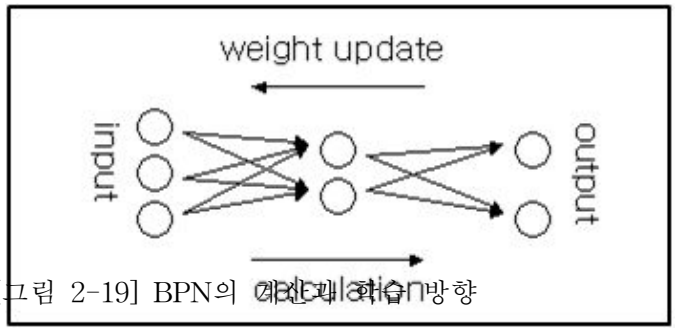
5-5. 역전파 알고리즘(Back Propagation Algorithm)

역전파 알고리즘은 다층(Multilayer)이고, Feedforward 신경망에서 사용되는 학습 알고리즘이며, 학습의 방법은 지도 학습(Supervised Learning)이다.

역전파 알고리즘의 학습은 전향단계와 후향단계로 이루어진다. 학습에서 중요한 것은 연결강도의 갱신인데 역전파 알고리즘은 후향단계에서 연결강도의 갱신이 이루어진다. 전향단계는 신경회로에 입력 패턴을 제시하고, 각 노드에 대해서 입력함수와 활성화 함수를 이용하여 출력을 산출하는 단계이다. 이 때 입력신호는 전방향 즉, 출력층 방향으로만 전달된다. 초기에 임의로 정한 연결강도에 의하여 출력층에서 출력값은 목표값과 당연히 일치하지 않고 큰 오차가 발생하게 된다. 후향단계는 목표출력과 실제출력의 차이를 계산하여 오차를 구하고, 이를 역방향 즉, 출력층에서 입력층으로 순서대로 층과 층 사이의 연결강도를 갱신하는 단계이다. 연결강도를 조절한 후 다시 입력을 넣어 계산하면 출력은 처음 시도 때보다 오차가 작은 값을 얻게 된다. 이를 반복하여 시스템이 안정될 때까지(총 오차의 합이 정해진 오차의 기준치에 도달할 때까지) 실행시켜서 원하는 값에 수렴시키는 것이다. 연결강도를 조정할 때 그 보정 값을 오차에 비례하게 해주는 것은 델타규칙(Delta Rule)에 LMS(Least Mean Square)학습 규칙을 적용하여 일반화된 델타규칙(Generalized Delta Rule)으로 행하게 된다. 즉, 다층 구조에서 계층마다 역방향으로 차례대로 이 방법을 적용한다.

학습을 하기 위해서는 입력 데이터와 원하는 출력(o) 데이터가 있어야 한다. 간단히 학습의 개념을 살펴보면, 입력이 신경망의 가중치와 곱하고 더하는 과정을 몇 번 반복하면 입력의 결과값인 출력(y)이 나온다. 이 때 출력(y)은 학습 데이터에서 주어진 원하는 출력(o)와 다르다. 결국, 신경망에서는 (y - o)만큼의 오차(e = y - o)가 발생하며, 오차에 비례하여 출력층의 가중치를 갱신하고, 그 다음 은닉층의 가중치를 갱신한다. 가중치를 갱신하는 방향이 신경망의 처리 방향과는 반대 방향이다(이런 이유로 역전파라고 함). 다시 말해, 신경망의 처리는 입력층 → 은닉층 → 출력층의 방향으로 진행되며, 가중치 갱신의 학습방향은 출력층 → 은닉층으로 진행된다. [그림 2-19]는 BPN의 계산과 학습 방향을 나타낸다.

Collection @ chosun



[그림 2-19] BPN의 계산과 학습 방향

역전파 알고리즘의 단계별 처리 알고리즘의 흐름은 아래의 [표 2-8]과 같다.

[표 2-8] 역전파 알고리즘의 단계별 처리

1단계	신경망에 입력 데이터를 입력 노드에 적용하고, 입력에 따른 출력을 계산한다.
2단계	입력에 따른 출력과 원하는 출력간의 오차를 계산한다.
3단계	오차를 줄이기 위해 가중치의 증감 여부를 결정한다.
4단계	각각의 가중치를 얼마나 변화시킬 것인가를 결정한다.
5단계	4단계에서 결정된 값으로 가중치를 갱신(변화)한다.
6단계	모든 학습 데이터에 대해 오차가 적정 수준으로 감소하기까지 1단계에서 5단계를 반복한다.

역전파 알고리즘의 단점은 국부 최소점(Local Minimizer)에 수렴한다는 것이다. 개선방법으로 은닉층 유닛(Unit)의 수를 증가시키거나, 연결강도를 조정하는데 있어서 이득항을 작게 하거나, 연결강도의 초기치를 임의의 값으로 설정하여 원하는 최종 목표 값에 도달할 때까지 학습 횟수를 반복하게 하는 방법이 적용된다. 이러한 방법을 적용하여 영상 분류와 영상 복원을 겸한 신경 회로망을 설계하고, Autoassociative Memory 기법을 이용하여 입력과 똑같은 출력을 얻는 신경 회로망을 구성하여 영상 복원 및 영상 분류의 신뢰도를 향상시키기 위해 피드백 개념을 도입할 수 있도록 하는 연구도 진행되었다[37].

5-6. RBF 신경망

신경망은 입력과 출력 공간 사이에 비선형적관계를 추출하기 때문에 유용한 분류기이다. RBF(Radial Basis Function) 신경망의 장점은 단순한 구조와 선형의 학

Collection @ chosun

습 알고리즘을 사용함으로써 학습속도가 빠르다[38].

가. RBF 신경망의 구조

RBF 신경망은 입력층 X, 은닉층 Y, 출력층 Z의 3층으로 구성된다. 각 층은 각각 I, J, K 개의 노드를 포함한다. 은닉층은 입력벡터 X의 가우시안 함수로 계산되어 하나의 RBF로 나타나고 각 출력층 노드는 은닉층 노드의 출력들의 가중치를 곱해서 합한 값이 된다. 입력벡터 X는 은닉층의 하나 또는 그 이상의 노드로 그룹화 된다. 이와 같이 일반적인 RBF의 식은 [식 2-2]과 같다.

(식 2-1)

$$V = \{a, r, f, h_1 \cdots h_7, s_1 \cdots s_{15}, d_1 \cdots d_{15}\}$$

(식 2-2)

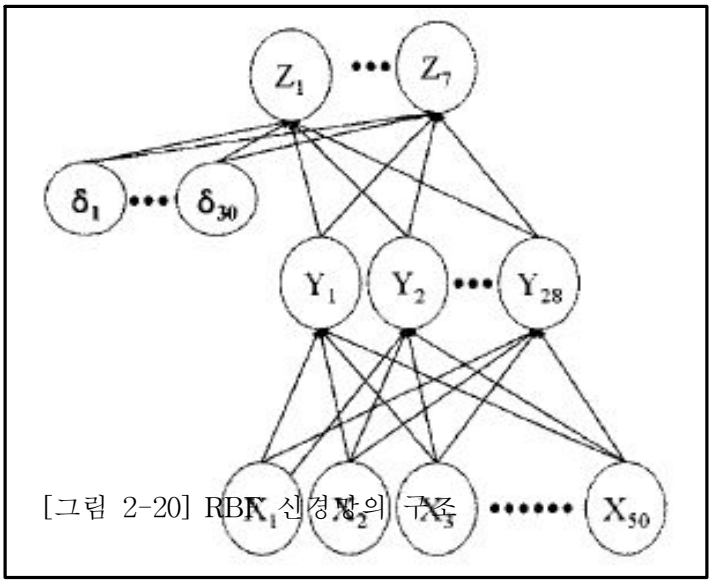
$$F_k(x_i) = \sum_{j=0}^{M} w_{jk}\phi_j(x_i)$$

$$\phi_j(x_i) = exp\left(-\frac{\| x_i - t_{ij} \|^2}{\sigma^2}\right)$$

여기서 j는 입력층의 노드 위치, k는 출력층의 노드 위치를 말한다. M은 입력 노드의 수로 (식 2-1)에서 구한 특징벡터 V에서 동적 특징( )을 제외한 특징들의 수로 여기서는 50이다. 이 때, 특징벡터 V는 0~1사이의 값으로 정규화된 후에 입력노드로 보내진다. t_{ij}과 는 각각 가우시안 RBF의 중앙값 벡터와 반경, w_{ik}는 입력층과 출력층사이의 연결강도 벡터, F_k는 k번째 출력 노드의 값을 나타낸다.

이질적인 은닉 노드의 분류를 위해 추가되는 특징 $_i$는 출력층으로 바로 연결된다. [그림 2-20]은 RBF 신경망의 구조를 나타낸다. 신경망의 입력값 X는 앞에서 구한 50개의 정적 특징이 사용되었고, 이는 K-means 알고리즘을 사용하여 28개의 은닉층 노드를 생성하였다. 여기에 30개의 동적 특징이 추가되어 최종 7개의 동작에 대한 값을 계산하게 된다.

Collection @ chosun



[그림 2-20] RBF 신경망의 구조

나. RBF 신경망의 학습

RBF 신경망의 은닉층을 구성하려면 은닉층 뉴런의 수를 결정해야 한다. 은닉층 뉴런을 결정하는 가장 단순하고 쉬운 방법은 각 학습패턴 마다 뉴런을 하나씩 할당하는 것이다. 그러나 대부분 실제 응용은 학습패턴의 수가 많고 입력차원의 수가 크므로 이 방법은 실용적이지 못하다. 그러므로 학습패턴을 K-means이나 LVQ 같은 군집화 알고리즘으로 적절한 군집을 찾아낸 후 각 군집마다 뉴런을 할당하는 것이 합리적이고 실용적이다[39]. 이러한 RBF 신경망에 K-means 알고리즘을 사용하여 은닉층을 구성하고, 정적 특징의 학습 데이터들을 학습하여 K를 구하여 가장 적절한 값으로 은닉층 노드수를 정하여 동적 특징의 출력층으로 바로 연결되어 가중치 역할을 한다[40].

5-7. LVQ 신경망

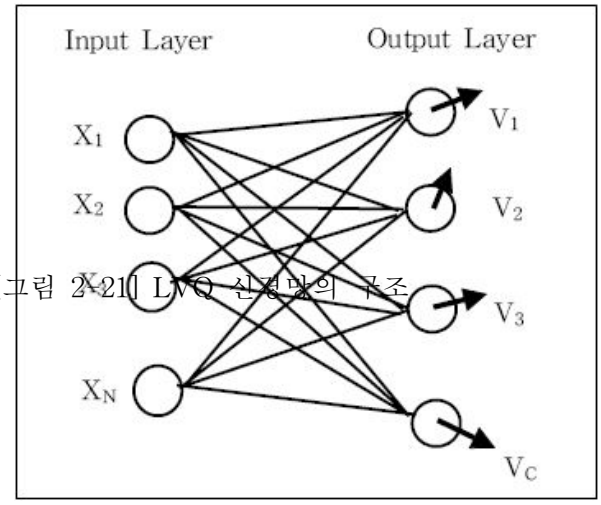
LVQ(Learning Vector Quantization) 신경망은 Kohonen이 제안한 알고리즘으로서[41] 클래스를 잘 구분할 수 있도록 경계를 형성하는 것을 목적으로 한다. LVQ는 교사학습과 비교사학습이 동시에 존재하는 형태로 구성되어 있으며, 연결 벡터를 원형 벡터로 하여 학습하는 알고리즘을 이용한다. LVQ의 학습은 입력과 출력 사이의 연결 벡터인 중심 벡터와 입력 벡터의 거리를 비교하여 중심 벡터와의 거리가 가장 가까운 입력 벡터가 승리 벡터가 되고, 중심 벡터와 승리 벡터가 같은

Collection @ chosun

클래스로 분류되면 연결 벡터를 증가시키고 다른 클래스로 분류되면 연결 벡터를 감소시킨다. 이러한 과정을 통해 입력되는 얼굴의 특징 벡터는 분류가 되고 연결 벡터는 학습하게 된다. 하지만 우수한 성능에도 불구하고 LVQ는 중심 벡터의 초기값을 선정하는 방법에 따라 성능이 좌우될 수 있다. LVQ의 중심 벡터는 클래스의 경계를 결정하고 그것의 수렴에 영향을 미치기 때문에 중심 벡터의 초기값을 선정하는 것은 매우 중요한 문제이다. 일반적으로 중심 벡터를 정하는 방법은 입력 패턴의 학습 테이터 중에서 임의로 선택하거나 또는 임의적(Random)으로 선택하는 방법이 사용되나 정확한 분류에는 적합하지 않다.

가. LVQ 신경망의 학습

LVQ 신경망은 입력층과 출력층으로만 구성된 단층의 신경망으로서 입력층의 뉴런은 출력층의 모든 뉴런들과 연결되어 있는 구조이다. 입력 벡터 x가 신경망에 입력되면 중심 벡터와 x의 유클리디언 거리를 기반으로 출력 뉴런들이 경쟁을 벌여 입력 벡터와의 거리가 가장 작은 한 개의 뉴런이 출력되며, 이 뉴런을 승리 뉴런이라고 한다. 뉴런에 클래스가 할당된 경우, 승리 뉴런의 클래스가 출력이 된다. LVQ는 경쟁시 신경망이 형성하는 결정 경계를 베이지언 최적 결정 경계에 접근하도록 고안된 방법으로 Kohonen은 교사학습 LVQ 알고리즘으로 LVQ1, LVQ2를 제안하였다[41]. [그림 2-21]은 LVQ 신경망의 구조이며, [표 2-9]는 LVQ 신경망의 알고리즘이다.



[그림 2-21] LVQ 신경망의 구조

Collection @ chosun

LVQ 신경망을 이용하여 최적의 클래스 중심 벡터를 찾기 위해 K-Means 클러스터링 알고리즘을 사용한 연구는 초기 클래스의 대표 벡터를 LVQ 신경망의 초기 가중치로 이용하며, 학습 데이터로부터 LVQ 신경망을 통해 최적의 클래스 대표 벡터를 얻은 후 학습을 통해 얼굴인식 시스템에 적용하였다[42].

[표 2-9] LVQ 알고리즘

1. 입력패턴 집합 $X=\{x_1, x_2, \ldots, x_n\}$, 참조패턴 집합 $V=\{v_1, v_2, \ldots, v_c\}$
   : 입력패턴 수, n : 클러스터 수
   T : 총 반복회수, Θ : 허용오차값 설정
2. $V_0=\{v_{1,0}, v_{2,0}, \ldots, v_{c,0}\}$: 0과 1사이의 난수로 초기 연결강도값 설정
   학습률 a_0 : 0과 1사이의 값 또는 지정된 Θ 값으로 초기학습률 설정
3. For t=1,2,3,...., T
   For k=1,2,3,...., n
   a. 승자 참조패턴 선택
   $x_k - v_{p,t} = \min\{x_k - v_{j,t}\}, (1 \le j \le c)$
   b. 승자 참조패턴 수정
   x_k와 $v_{p,t}$가 같은 부류일 때, $v_{p,t+1}=v_{p,t}+a_t(x_k-v_{p,t})$
   x_k와 $v_{p,t}$가 다른 부류일 때, $v_{p,t+1}=v_{p,t}-a_t(x_k-v_{p,t})$
   c. next
4. 반복에 따른 오차 계산
   $E_t = V_t - V_{t-1} = \sum_k \sum V_{k,t} - V_{k,t-1}$
5. 만약, $E_t \le \Theta$ 이면 학습 중단, 아니면 학습률 조정 next t

Collection @ chosun

III. 피부색 정보를 이용한 손 영역 검출

컬러 공간에서 사람의 피부색 분포는 자연계의 다른 물체와 구분할 수 있는 뚜렷한 특징을 가지고 있다. 비록 사람과 인종마다 피부색은 다르지만 색차 평면상에서의 그 분포는 매우 비슷하면서도 협소한 영역을 차지하고 있고, 단지 밝기(Intensity)의 차이만이 있어, 색차 성분을 이용하면 쉽게 피부색 영역을 추출할 수 있다[44]. 색상 정보를 이용하여 일반 영상에서 손을 분리해내기 위해서는 적합한 Color Space에서 피부색을 가장 잘 모델링 할 수 있는 것이 필요하다. 인간의 피부색은 RGB좌표계에서 아주 좁은 영역에 분포한다.

1. Skin-Tone에 의한 세그멘테이션

피부색 정보로 세그멘테이션 될 영상은 전경과 배경으로 분리된 후 사람의 손일 가능성이 높은 전경 영역에 대해서만 처리가 이루어지도록 하였다. 영상이 전경과 배경으로 이루어져 있을 경우 대부분의 의미 있는 정보는 전경 영역에 집중하게 된다. 단순히 전경과 배경을 분리하는 것은 의미가 없고 불필요한 배경으로부터 관심의 대상인 전경 영역만을 추출하는 것이다. 전·배경 분리 과정을 영상 인식, 감시 시스템 등의 전처리로 추가할 경우 정보가 집중된 전경 영역만을 이용하여 처리 하게 됨으로써 전체 시스템의 성능과 정확성을 향상시킬 수 있게 된다. 영상을 전경과 배경으로 분리하기 위해서는 전경이 없는 영상만으로 배경 모델을 생성한 후 새로운 영상과 배경 모델과의 차이를 전경으로 판단할 수 있다.

Blue-screen과 같은 물리적 제약이 없이, 영상만을 이용하여 전·배경 분리를 하는 방법은 크게 Intensity 기반, 깊이 기반, Motion 기반의 방법으로 나눠 볼 수 있다. Intensity 정보를 이용한 방법은 일정한 시간의 영상들로부터 배경 모델을 생성한 후, 새로운 입력과 배경 모델의 Intensity를 비교하여 그 유사한 정도가 기준치 이하이면 전경, 이상이면 배경으로 구분하여 배경을 제거하는 방법이다[26][45]. 이 방법은 실시간 처리가 가능하고 쉽게 구현이 가능하지만 그림자와 조명 변화 등 주위환경에 민감한 단점을 지니고 있다. 조명변화에 의한 오류를 줄이기 위해 명암

Collection @ chosun

과 색차 신호를 분리하여 처리하는 방법 등이 제안되었으나, 실제 실험 결과 정도의 차이를 보일 뿐 만족할 만한 결과를 얻을 수 없었다. 깊이 정보를 이용한 배경 제거 방법은 영상으로부터 깊이 정보를 구한 후 깊이의 정도에 따라 카메라에 가까운 물체는 전경으로 상대적으로 먼 물체는 배경으로 분할하는 방법과, 새로운 영상의 깊이 정보와 미리 계산된 배경 모델의 깊이 정보를 비교하여 변화가 있는 부분을 전경으로 보는 방법[45] 등이 있다. 이 방법은 조명 변화와 같은 주위 환경의 변화에 영향을 받지는 않지만, 계산량이 많아 처리시간이 오래 걸린다. Motion 정보를 이용한 배경 제거 방법은 배경이 고정된 상황에서 움직이는 물체를 전경으로 추출하는 방법으로 물체의 움직임이 큰 경우에 오류가 발생되고, 잡음에 민감하여 전경과 배경이 함께 움직일 경우 분할에 실패하게 된다.

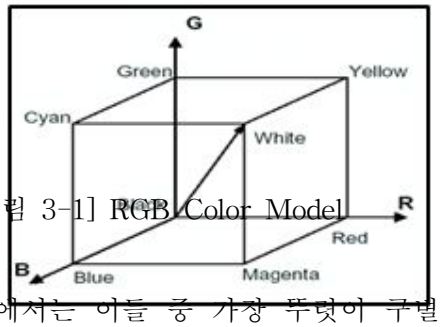
본 논문에서는 손의 특징 점(Hand Feature Point)의 추출을 위해서 여러 가지 색체모델 중 YUV Color Model을 이용한 Skin Tone 추출 방법을 사용하였다. 물론 이 방법은 배경이나 의상에 스킨 톤이 포함되어 있지 않다는 가정을 포함하고 있다.

RGB 색상 모델은 서로 가산될 수 있는 삼원색인 빨강(Red), 녹색(Green), 그리고 파랑(Blue)으로 구성된다. 이들 색상의 분광 요소들이 부가적으로 복합되어 색상을 만들어 낸다. RGB 색상 모델은 [그림 3-1]과 같이 각 축의 모서리가 빨강, 녹색, 그리고 파랑인 3차원의 입방체로 표현된다. 검정색은 원점이며, 흰색은 입방체의 반대쪽 끝이다. Gray-scale은 검정색에서 흰색으로 이어지는 점선으로 표현된다. RGB 색상 모델은 컴퓨터 그래픽스 시스템의 설계를 간단하게 하지만, 모든 어플리케이션에 이상적이지 않다. 빨강, 녹색, 파랑의 각 색상 요소들의 상관관계는 너무 크기 때문에 몇몇 영상 처리 알고리즘의 수행이 어렵게 된다. 이에 반해 YUV 색상 모델은 색상 정보로부터 광도를 분리하는 또 다른 색상 모델이다. 영상의 광도는 Y로 표시되고, 영상의 푸른 정도와 붉은 정도는 UV로 기호화된다. 따라서 YUV 색상 모델은 [그림 3-2]와 같이 Y를 제외한 UV의 2차원 직교좌표계로 표시될 수 있다. 또한 YUV 색상 모델은 PAL이나 SECAM과 같은 컬러 영상의 표준으로 사용되며, RGB 정보로부터 YUV 모델로의 변환은 (식 3-1)과 같이 구할 수 있다.

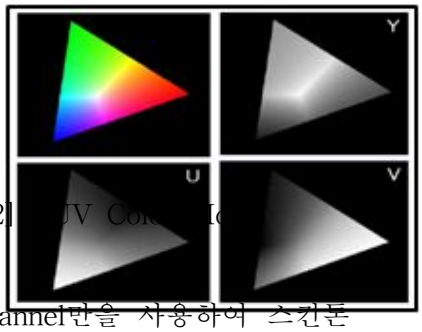
Collection @ chosun

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad \text{(식 3-1)}$$

특히, YUV 색상 모델에서 일반적인 피부색상은 전체 색상 영역 중 매우 좁은 영역에 분포되므로 다른 색상 모델에 더욱 효율적으로 피부색을 추출할 수 있다. 또한 YUV 색상 모델의 특성상 RGB 색상 모델에 비해 조명 변화에 덜 민감하기 때문에 조명 조건의 변화에도 안정적인 결과를 보인다.



[그림 3-1] RGB Color Model



[그림 3-2] YUV Color Model

본 논문에서는 이들 중 가장 뚜렷이 구별되는 U Channel만을 사용하여 스킨톤을 추출하였다. Color Segmentation 시 Skin Tone에 해당하는 Pixel 값들을 임의의 확률 모델로 모델링하고, 이에 맞는 Segmentation을 수행해야 하지만, U Channel 내의 Skin Tone이 다른 영역과 뚜렷이 구별되며, 그 분포영역 또한 좁기 때문에 단순히 임계값(Threshold)만을 취하여 영역을 추출하였다. 또한 추출한 영역에 대해 Morphological Operation인 Opening과 Closing을 수행하여, 영향력이 작은 Blob을 제거하여 사용한다. 그러나 조명변화가 심하거나 일반 배경 안에서는 YUV 컬러 모델을 사용하는 방법은 실시간으로 손 영역을 추출하는데 속도차원에서 많은 계산 시간을 요구하게 된다. 그래서 스킨 컬러 필터를 적용하여 스킨 컬러 영역을 추출하였다. [그림 3-3]은 입력영상에 피부색 영역을 추출한 영상이다.

Collection @ chosun

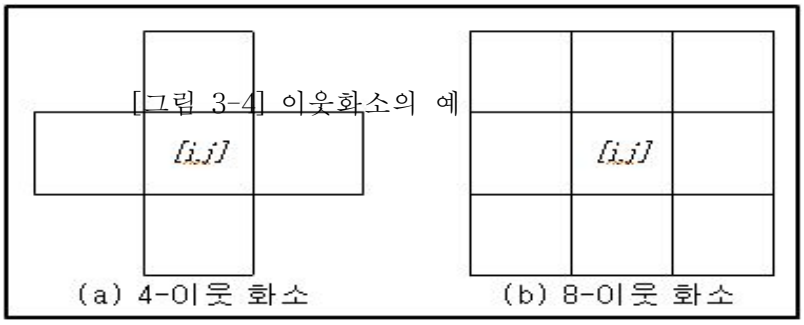
[그림 3-3] 입력 영상의 손 영역 추출

2. 손 영역 추출

입력 영상에서 피부색 정보를 이용하여 손 영역을 추출할 때 컬러 필터를 사용하더라도 손만을 정확히 추출하기란 쉽지 않다. 본 연구는 컬러 필터가 적용되고 남아 있는 피부색 정보의 픽셀들은 대부분 손에 속해 있기 때문에 횡 좌표 X축으로 피부색 픽셀들의 수를 계산하여 배경영역으로부터 손 영역만을 분리한다(Labeling).

2-1. 레이블링(Labeling)

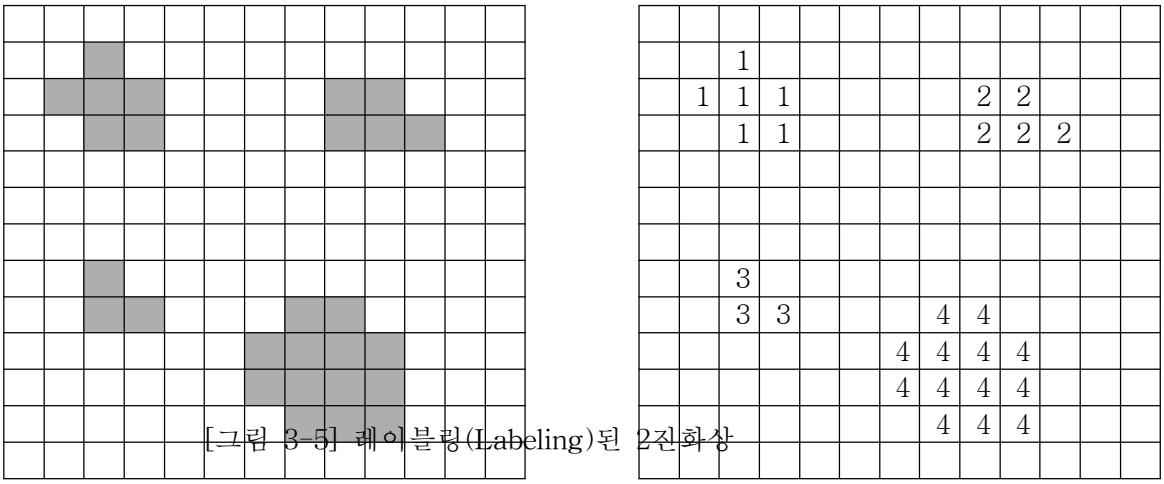
물체의 위치나 방향 외에 각각의 물체를 구분하기 위하여, 화소들이 서로 연결되어 있는가, 또는 떨어져 있는가를 알아야 한다. 한 화소가 다른 화소들에 연결되어 있는지의 여부를 알기 위하여, [그림 3-4]에서와 같이 4-이웃화소(Neighbor)와 8-이웃화소를 정의한다.



[그림 3-4] 이웃화소의 예

Collection @ chosun

[i, j] 위치에 있는 화소의 4-이웃화소는 [i+1, j], [i-1, j], [i, j+1], [i, j-1]에 있는 화소들로 구성되며, 8-이웃화소는 4-이웃화소 및 [i+1, j+1], [i+1, j-1], [i-1, j+1], [i-1, j-1]에 있는 화소들로 구성된다. 이 [i, j]위치에 있는 화소의 4-이웃화소를 4-연결도(Connectivity)의 이웃화소들이라 하고, 8-이웃화소를 8-연결도의 이웃화소들이라고 한다.



[그림 3-5] 레이블링(Labeling)된 2진화상

(a) 원래의 2진화상 (b) 레이블링된 2진화상

일반적으로, 화상 배열에서 서로 이웃하여 있는 화소들의 집합을 연결성분(Connected Component)이라고 부른다. 머신비전에서 가장 흔한 연산중의 하나는 주어진 화상에서 이러한 연결성분들을 찾는 것이다. 연결성분들에 속한 화소들은 어떠한 물체를 나타낼 가능성이 크기 때문이다. 한 연결성분에 같은 레이블(번호; 정수 값)을 붙이고, 다른 연결성분에는 다른 번호를 붙이는 처리를 레이블링이라고 한다. 2진화상의 레이블링의 예가 [그림 3-5]에 보여진다.

[그림 3-5](a)는 임계값처리를 통해 얻어진 2진화상으로서, 어두운 색으로 처리된 화소들은 1의 값을 가지며, 이외의 화소들은 0의 값을 가지며, 4개의 연결성분들이 있다. [그림 3-5](b)는 적절한 레이블링 과정을 통해 각각의 연결성분이 1, 2, 3, 4의 레이블을 갖는 화상을 보여준다. 이렇게 레이블링된 화상으로부터 얻어진 4개의 연결성분들은 각각의 어떤 물체를 나타낼 수 있는 가능성이 있으며, 각각의 연결성분에 대해 크기, 위치, 방향 등의 특징들을 계산할 수 있다.

Collection @ chosun

화상에 있는 모든 연결성분들을 찾고, 같은 연결성분에 존재하는 모든 화소들에 하나의 고유한 레이블을 붙여주는 알고리즘을 성분 레이블링 알고리즘(Component Labeling Algorithm)이라고 하며, 회귀(Recursive)와 순차(Sequential) 알고리즘도 있다. 회귀 알고리즘은 직렬 컴퓨팅에서는 그 계산 시간이 오래 걸리므로, 병렬 컴퓨터에서 많이 사용된다. 순차 알고리즘은 회귀 알고리즘에 비해 계산시간이 짧고 메모리가 적게 들며, 주어진 화상에 대해 2번의 전체 조사(scanning)만으로 계산이 끝난다. 여기서는 순차 알고리즘만을 설명한다.

▣ 4-연결도를 사용한 순차 연결성분 알고리즘(Sequential Connected Components Algorithm Using 4-Connectivity)

1) 왼쪽에서 오른쪽으로, 위에서 아래로 화상을 조사한다.
2) 만약 [i, j] 화소가 1값을 가지면, 위쪽 ([i-1, j])과 왼쪽 ([i, j-1]) 이웃화소들에 대해,
   ① 두 이웃화소들 중에서 하나만이 레이블을 가지고 있으면, 그 레이블을 [i, j] 화소에 부여한다.
   ② 두 이웃화소들이 똑같은 레이블을 가지고 있다면, 그 레이블을 [i, j] 화소에 부여 한다.
   ③ 두 이웃화소들이 서로 다른 레이블을 가지고 있다면, 위쪽 이웃화소의 레이블을 부여하고, 준비된 등가 테이블에 두 레이블을 등가레이블로 기록한다.
   ④ 두 이웃화소들이 레이블을 가지고 있지 않다면, 새로운 레이블을 부여하고, 등가 테이블에 이 레이블을 기록한다.
3) 레이블이 부여되지 않은 화소들이 남아 있으면, 단계 2)로 간다.
4) 등가 테이블에서 각각의 등가 집합에 대해 가장 낮은 레이블을 찾는다.
5) 화상을 조사하여, 각 레이블을 등가 집합에서의 가장 낮은 레이블로 바꾼다.

여기서 등가 레이블이라 함은 같은 연결 성분들에 대한 서로 다른 레이블들을 말 하며, 이 등가 레이블들의 집합을 등가 집합이라고 한다.

이 알고리즘은 전체 화상에 대해 두 번의 조사 과정이 필요하다. 첫 번째 조사에서는 부분 영역들을 얻고 부분 영역들 간의 연결성을 알아낸다. : 과정 1), 2), 3) 두 번째 조사에서는 등가인 레이블들을 검사하여 다시 레이블링을 한다. 위의 과정

Collection @ chosun

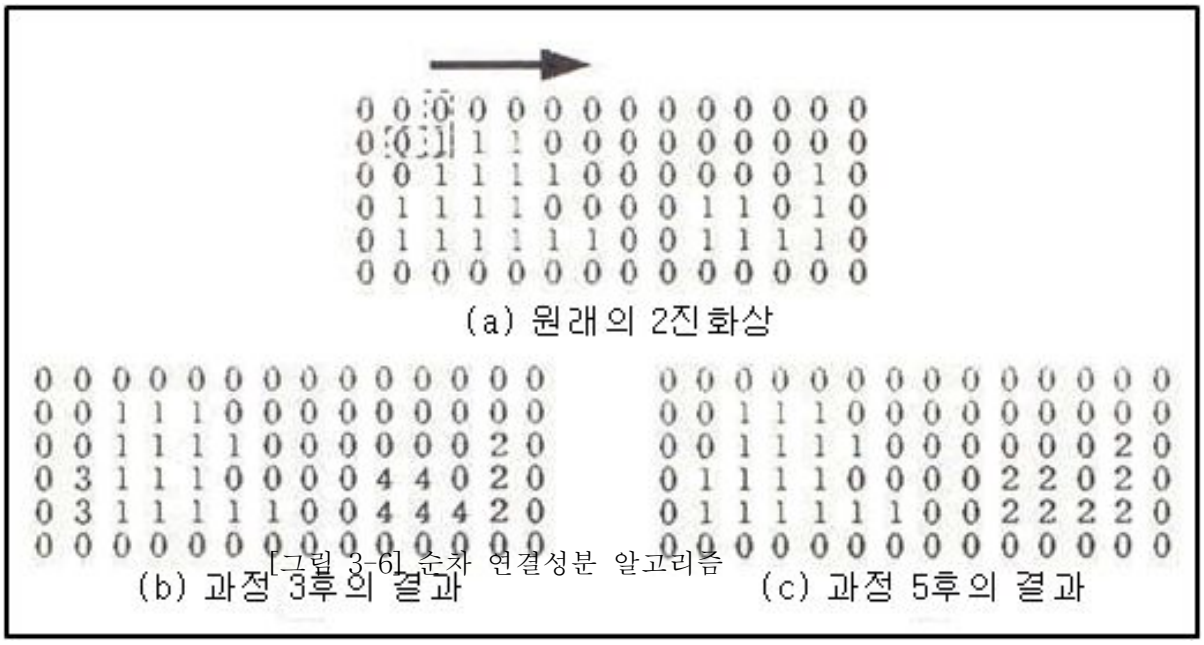
을 Pseudo 코드로 나나내면 아래의 [표 3-1]과 같다.

[표 3-1] Pseudo 코드

```
k=1;
if p[r][c] = 1
{
   if (p[r-1][c] = 1 && p[r][c-1] = 0)
      label[r][c] = label[r-1][c];
   if (p[r-1][c] = 0 && p[r][c-1] = 1)
      label[r][c] = label[r][c-1];
   if (p[r-1][c] = 1 && p[r][c-1] = 1)
      label[r][c] = label[r-1][c];
   if (p[r-1][c] = 0 && p[r][c-1] = 0)
   {
      label[r][c] = k;
      k = k + 1;
   }
}
```

순차 연결성분 알고리즘의 예를 [그림 3-6]에 나타내었다. 1), 2), 3)의 과정을 통해 얻어진 등가 테이블은 ET ={ }이고, 각각의 등가 집합은 ={1,3}, ={2,4}이다. 즉, 등가 테이블 ET 는 2개의 등가 집합 , 로 구성되고, 은 등가 레이블들의 집합으로서 라벨 1과 3이 서로 등가 레이블임을 뜻한다. 마찬가지로 의 레이블 2와 4는 서로 등가 레이블이다. 4)의 과정을 통해 각 등가 집합 ES_1, ES_2 에서 가장 낮은 레이블은 각각 1과 2이다. 5)의 과정을 거치면, 레이블 3을 갖고 있던 화소들은 모두 레이블 1로 바뀌게 되고, 레이블 4를 가지고 있던 화소들은 모두 레이블 2로 바뀐다. 따라서 레이블링 알고리즘을 적용한 뒤의 화소들은 레이블 1,2 중의 하나를 갖게 되고 2개의 연결성분이 얻어진다.

Collection @ chosun



[그림 3-6] 순차 연결성분 알고리즘

[그림 3-3](a)의 입력영상처럼 전경과 배경 입력 영상 중에서 피부색 정보를 포함하는 것은 대부분이 손이기 때문에 Labeling 알고리즘을 이용하여 필터링된 픽셀들의 가장 높은 계수를 가진 쪽을 손으로 귀착시킨다. 그러나 특별한 조명을 이용하지 않기 때문에 각 영상 마다 잡음과 손 추출 시 손의 미묘한 잡음들이 많이 포함되어 있다. 이를 해결하기 위해 Mathematical Morphology에 Closing을 이용하여 이런 문제를 해결했다. 이 알고리즘은 다이나믹 상황에서 매우 좋은 결과를 산출할 수 있다.

2-2. Mathematical Morphology

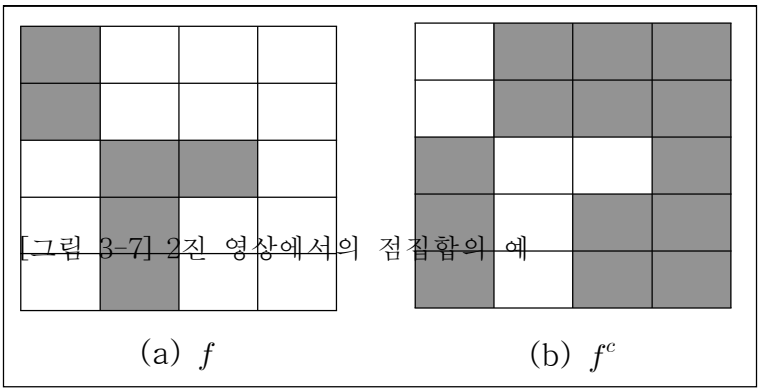
수학적 형상학은 영상 내에 존재하는 객체의 모양과 구조를 추출하여, 영상을 분석하고 처리하는데 사용되는 이론으로서, 동물이나 실물의 형태와 구조를 다루는 생물학의 한 분야인 형상학을, 영상처리 및 분석에 적용될 수 있도록 수학적 이론 위에 재구성되었다.

수학적 형상학(또는 형상학)은 앞으로 정의될 형상학적 연산을 적절히 조합하여 사용함으로써, 다양한 필터링 기술과 Thinning, Pruning과 같은 복잡한 문제를 해결할 수 있으며, 더 나아가 영상에 존재하는 객체를 각각 분리해 내거나 (Segmentation) 혹은 객체의 특징 추출 및 측정, 패턴 인식 등에 사용되고 있다.

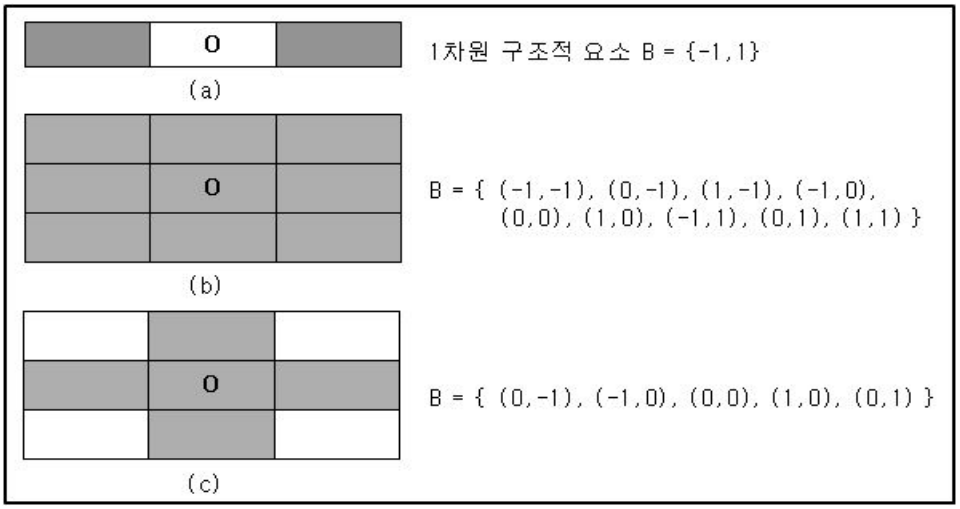
Collection @ chosun

가. 형상학의 기초

2진 영상의 경우를 보면, 2진 영상은 이차원의 점집합으로 표현할 수 있다. 집합 f를 영상에서의 객체 내부에 포함되는 점들의 집합이라고 하고, 이 점들이 각각 1의 값을 갖는다고 하면, 여집합 f^c 에서는 영상 내의 배경이 1의 값을 갖게 된다. O로 표시되는 원점은 좌표(0,0)를 가지며, 임의의 점의 좌표는 (x,y)로 표현될 수 있다. [그림 3-7]은 집합 f = { (1,0), (1,1), (1,2), (2,2), (0,3), (0,4) }에 해당하는 2진 영상의 예를 보여준다. 객체에 해당하는 점들은 f 로 나타내었다. 이때, 형상학적 변환 Ψ 는 점집합 f 로 표현되는 영상과 구조적 요소(Structuring Element)라 부르는 또 다른 작은 점집합 B관계로 주어진다. 이때 구조적 요소 B는 자신의 원점 O를 기준으로 표현된다. [그림 3-8](a)에서는 원점 O가 구조적 요소 B의 원소가 아닐 수도 있다는 것을 보여주고 있다.



[그림 3-7] 2진 영상에서의 점집합의 예

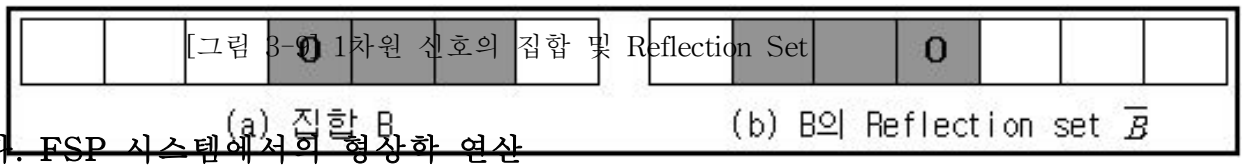


Collection @ chosun

[그림 3-8] 전형적인 구조적 요소의 몇 가지 예

나. 형상학의 연산

형상학 연산을 이해하기 위하여 먼저 함수와 집합(Set)에 대해서 알아야 한다. 함수는 그레이스케일 입력 신호를 나타내며, 1차원 신호일 경우에는 $f(n)$으로, 2차원 신호에서는 $f(m,n)$으로 표현된다. 여기에서 $f(n)$은 1차원 신호 f 상에서의 n번째 값을 말하고, $f(m,n)$은 2차원 신호 f의 (m,n)번째 값을 나타낸다. 특히 2차원 입력 함수 f는 그레이 스케일 영상을 나타낼 수 있다. 한편 집합은 유클리드 공간에서 점들의 모임으로서 값을 가지지 않고, 단지 원점으로부터의 위치에 대한 정보만을 나타낸다. [그림 3-9]에서는 구조적 요소 B를 집합으로 나타낸 예를 보여주고 있다 이때, 원점에 대칭인 집합을 Reflection Set이라 하고 $\overline{B}$로 나타낸다. 즉, [그림 3-9](a)의 구조적 요소를 B={0,1,2}라 하고 원점이 원소 0의 위치에 있다고 할 때 Reflection Set은 $\overline{B}$= {0,-1,-2}가된다[그림 3-9](b)).



[그림 3-9] 1차원 신호의 집합 및 Reflection Set

다. FSP 시스템에서의 형상학 연산

FSP Dilation 와 FSP Erosion 는 각각 $f \oplus B$, $f \ominus B$로 표기하여 (식 3-2),(식 3-3)과 같이 정의 된다.

$$g_d(m,n) = (f \oplus B)(m,n) = Max_{(i,j)\in B} f(m-i, n-j) \quad \text{(식 3-2)}$$

$$g_e(m,n) = (f \ominus B)(m,n) = Min_{(i,j)\in B} f(m+i, n+j) \quad \text{(식 3-3)}$$

이때 구조적 요소 B는 함수가 아니라 집합으로 표현된다. [그림 3-9](a)의 구조적 요소를 가지는 경우를 생각해 보면, 연산의 결과가 되고 erosion 연산의 출력은 가

$$g_d(m,n) = Max f\{(n), f(n-1), f(n-2)\}$$

$$g_e(m,n) = Min f(n), f\{(n+1), f(n+2)\}$$

Collection @ chosun

된다.

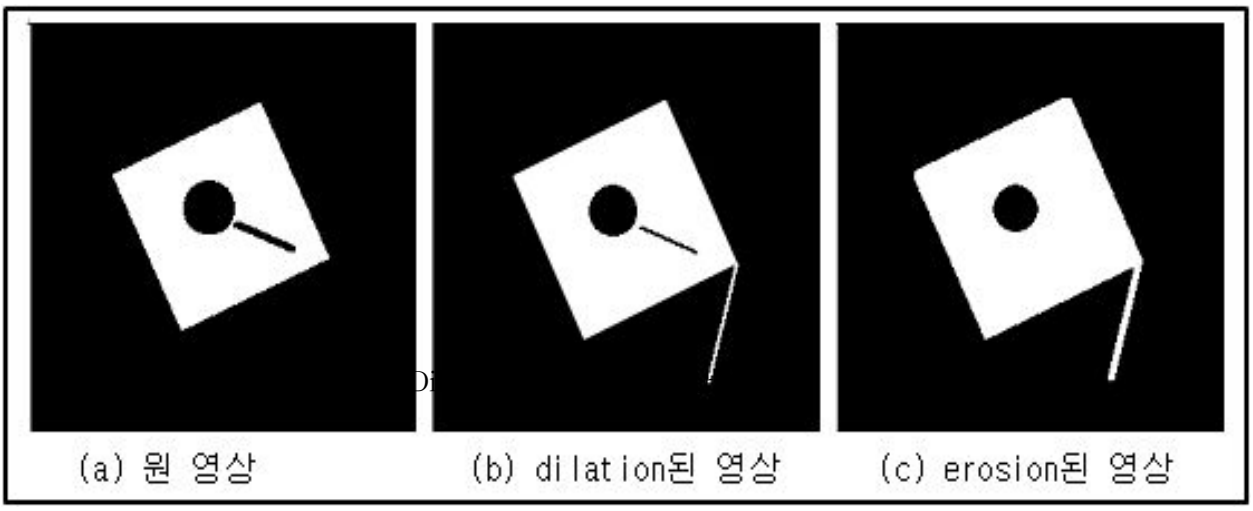
2차원의 경우에도 1차원의 경우와 마찬가지로 출력을 구할 수 있으며, (식 3-2),(식 3-3)을 적용하면 결과는 각각 (식 3-4),(식 3-5)이 된다.

(식 3-4)

$$g_d(m,n) = Max\{f(m,n), f(m-1,n), f(m,n-1)\}$$

(식 3-5)

$$g_d(m,n) = Min\{f(m,n), f(m+1,n), f(m,n+1)\}$$



[그림 3-11] 실제 영상에 적용한 FSP Dilation과 FSP Erosion 연산의 예

Collection @ chosun

[그림 3-10]에서는 둥근 공 모양을 가진 구조적 요소를 사용하여 Dilation과 Erosion 연산의 효과를 직관적으로 확인할 수 있는 예를 보여주고 있다. [그림 3-10](b)의 Dilation 연산에서 구조적 요소는 전체 입력 영상에 대해서 좌에서 우로, 위에서 아래로 차례로 움직이면서 매 순간마다 구조적 요소 안에 포함되는 값들 중 최대값을 취하게 된다. 이렇게 Max 연산을 통하여 큰 값을 나타내는 밝은 객체 부분이 더 커지는 것을 알 수 있다. Erosion 연산은 그 반대로 Min 연산을 취하기 때문에 어두운 부분이 확장되었다는 사실을 알 수 있다. [그림 3-11]에서는 실제 그레이스케일 영상에 대해서 Dilation과 Erosion을 적용한 예를 보여주고 있다.

라. SP 시스템에서의 형상학 연산

0과 1의 값을 갖는 2진 입력 신호는 유클리드 공간에서 집합으로 취급할 수 있다. 따라서 2진 영상에 대한 형상학 연산은 SP 시스템으로 분류할 수 있다. SP Dilation은 FSP Dilation의 특수한 경우로서, 0과 1로만 이루어진 입력 신호의 Max 연산은 주어진 구조적 요소 내에 적어도 하나 이상의 입력 신호가 1이면, 1을 출력하기 때문에, Boolean OR 연산으로 대체될 수 있다. 이 성질을 이용하여 SP Dilation은 (식 3-6)와 같이 정의 될 수 있다.

$$g_d(m,n) = (f \oplus B)(m,n) = \bigcup_{(i,j)\in B} f(m-i, n-j)$$

(식 3-6)

여기서 $\bigcup$는 2진 도메인에서의 OR 연산을 나타낸다.

같은 방식으로 SP Erosion은 입력의 최소값을 구하게 되는데, 0과 1의 2진 입력의 경우에는 어느 한 입력이라도 0값을 가지면 그 출력은 0이 되고, 이는 AND 연산으로 나타낼 수 있다. 따라서 SP Erosion은 (식 3-7)과 같이 정의 할 수 있다.

$$g_d(m,n) = (f \oplus B)(m,n) = \bigcap_{(i,j)\in B} f(m+i, n+j)$$

(식 3-7)

여기에서 $\bigcap$는 2진 도메인에서의 AND 연산을 나타낸다.

SP 시스템에서의 형상학 연산은 OR 와 AND 논리 연산으로 대체 될 수 있기 때문에, 빠른 처리 속도의 하드웨어를 쉽게 구현할 수 있게 되었다. 2진 입력 영상

Collection @ chosun

에 대한 SP 시스템의 이러한 장점을 일반적으로 많이 사용되는 그레이스케일 영상에 적용하기 위해서 그레이스케일의 정보를 적절한 2진 정보로 대체해 사용할 수 있는 방법을 모색하게 되었다. 다시 말해 최대값이나 최소값을 구하기 위해, 끊임없이 비교 연산을 수행해야하는 FSP 연산을 논리 회로로 이루어진 SP 시스템을 이용하여 빠르게 처리할 수 있는 방법을 찾고자 한 것이다.

마. Opening과 Closing 연산

앞 절에 설명한 Dilation과 Erosion을 기본으로 하여 여러 가지 유용한 형상학적 연산들이 정의 된다. 이 중에서 가장 쉽게 얻어지는 형태로는 Dilation과 Erosion 연산을 차례대로 한 번씩 적용하여 얻어지는 Opening과 Closing 연산이 있다. 이때, Erosion 연산 후에 Dilation을 하는 것을 Opening 연산이라 한다. 입력 영상 f에 대하여 구조적 요소 B에 의한 Opening 연산은 로 표기하고, (식 3-8)과 같이 정의된다.

$$f \circ B$$

(식 3-8)

그레이스케일 입력에 대한 FSP Opening은 Erosion 결과에 대하여 Dilation 연산을 수행하여 얻어지며, (식 3-9)와 같이 표기할 수 있다.

$$g(m,n) = (f \circ B)(m,n) = [(f \ominus B) \oplus B](m,n)$$

(식 3-9)

2진 입력 영상에 대한 SP Opening은 Min, Max 연산을 AND, OR 연산으로 대체하면 다음 (식 3-10)으로 계산할 수 있다.

$$g_o(m,n) = Max_{(i,j)\in B}\{g_e(m-i,n-j)\}$$
$$= Max_{(i,j)\in B}\{Min_{(i',j')\in B}\, f(m-i+i',n-j+j')\}$$

(식 3-10)

$$g_o(m,n) = \bigcup_{(i,j)\in B} g_e(m-i,n-j)$$
$$= \bigcup_{(i,j)\in B} \bigcap_{(i',j')\in B} f(m-i+i',n-j+j')$$

Collection @ chosun

Opening과 반대로 Dilation 후에 Erosion을 하는 것은 Closing이라 부른다. 영상에 대하여 구조적 요소 B에 의한 Closing 연산은 $f \bullet B$ 로 표기하고, (식 3-11)과 같이 정의된다.

(식 3-11)

$$g_c(m,n) = (f \bullet B)(m,n) = [(f \oplus B) \ominus B](m,n)$$

Opening의 경우와 마찬가지로 그레이스케일 및 2진 입력 신호에 대하여 FSP Closing 과 SP Closing 연산을 각각 (식 3-12),(식 3-13)과 같이 정의 할 수 있다.

(식 3-12)

$$g_c(m,n) = min_{(i,j)\in B}\{g_d(m+i,n+j)\}$$

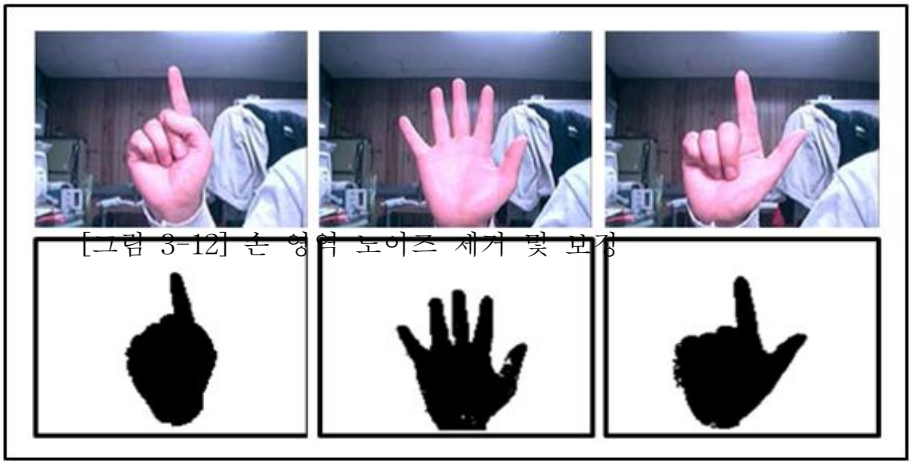
$$= min_{(i,j)\in B}\{max_{(i',j')\in B} f(m+i-i',n+j-j')\}$$

(식 3-13)

$$g_c(m,n) = \bigcap_{(i,j)\in B} g_d(m+i,n+j)$$

$$= \bigcap_{(i,j)\in B} \bigcup_{(i',j')\in B} f(m+i-i',n+j-j')$$

여기서 본 논문에서는 손을 추출한 영상에서 많은 노이즈와 손 형상이 매끄럽지 않은 형상을 발견하였기 때문에 개선하기 위해 (식 3-11)을 이용하여 [그림 3-12]처럼 손의 노이즈 제거 및 부드러운 손 형상을 얻을 수 있었다.



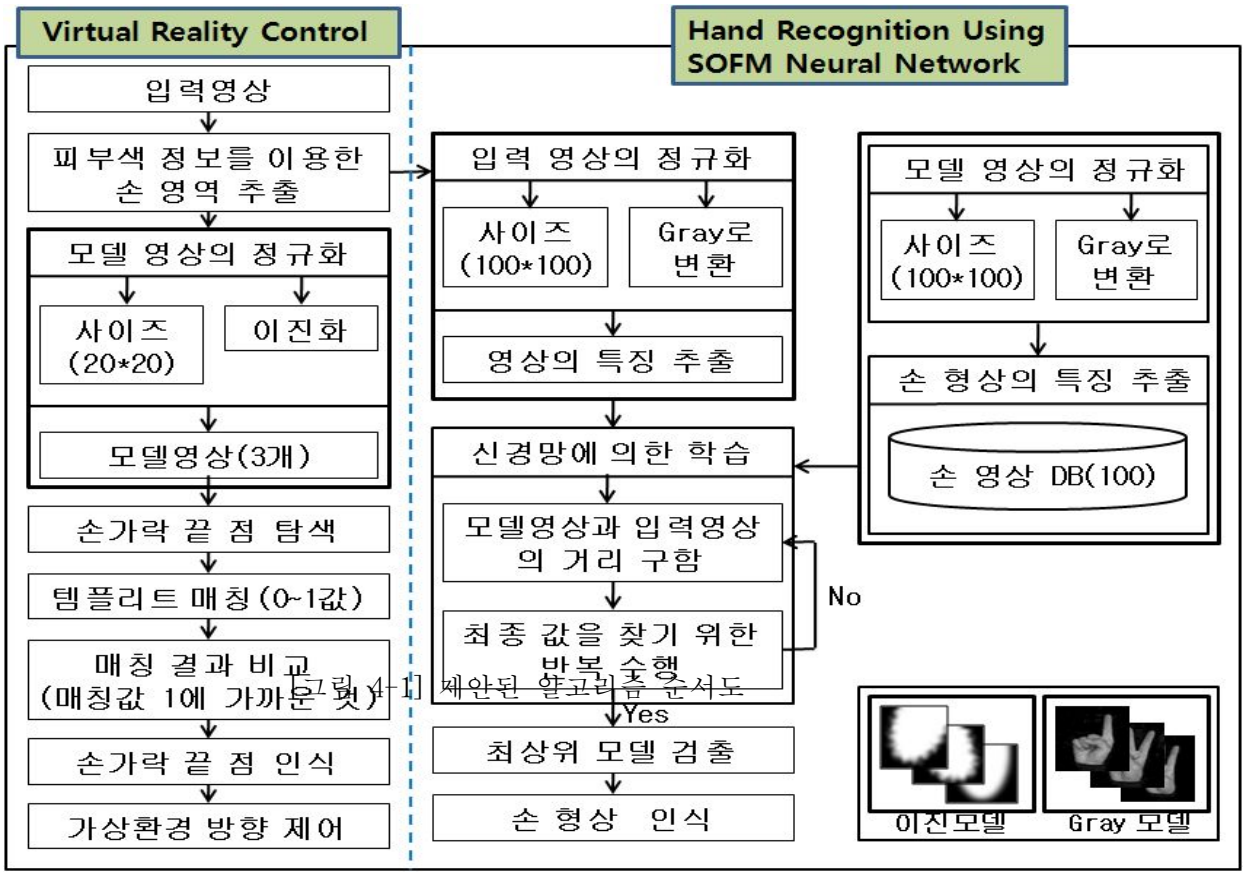
[그림 3-12] 손 영역 노이즈 제거 및 보정

Collection @ chosun

IV. 신경망을 이용한 손 형상 인식

1. 제안된 알고리즘 순서도

본 논문에서는 손 형상을 정확히 분석하고 인식하기 위해서 자기 조직화 특징 지도(SOFM:Self Organized Feature Map)라는 신경망 알고리즘을 사용한다[43]. 카메라로부터 들어오는 영상을 입력 영상으로 하여 영상의 전체영역에서 피부색 정보를 이용하여 손 영역만을 분리한 후 손 형상을 인식한다. 분리된 손 영역의 손가락 정보를 이용하여 손가락 끝 점을 탐색하고, 탐색되어진 손가락 끝 점을 템플리트 매칭을 이용하여 가상환경 상에서 전,후,좌,우와 같은 방향을 제어한다. 또한 분리된 손 영역을 손 형상으로 인식하기 위해 SOFM(Self Organized Feature Map) 신경망 알고리즘을 이용하였다. 본 논문에서 제안된 알고리즘 순서도는 [그림 4-1]에서 볼 수 있다.



[그림 4-1] 제안된 알고리즘 순서도

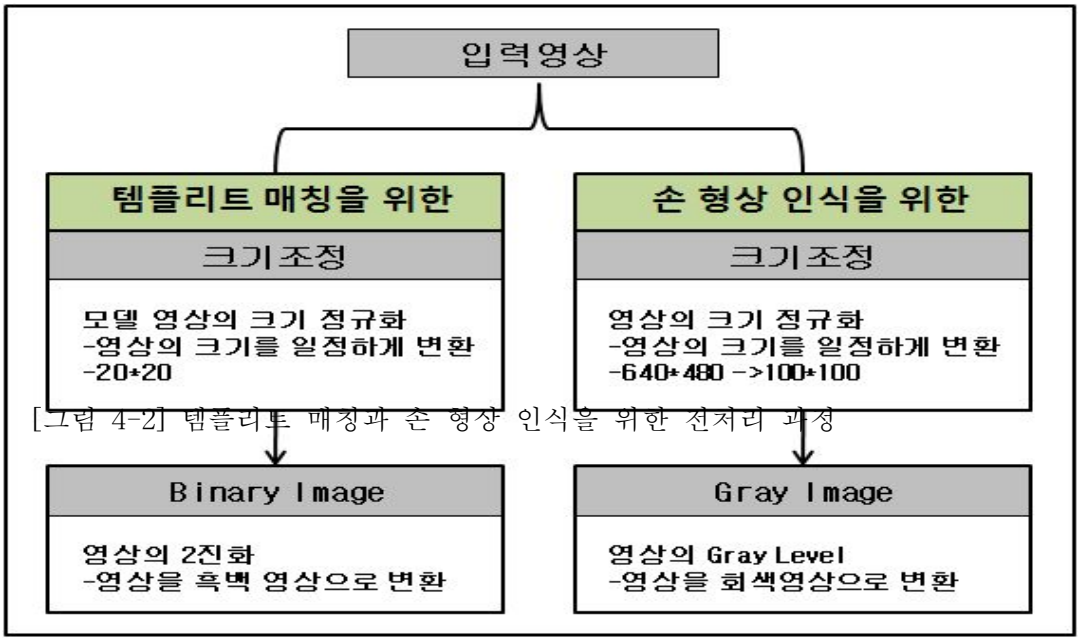
Collection @ chosun

[그림 4-1]의 "Virtual Reality Control"의 경우에는 3개의 모델영상으로 템플리트 매칭을 시도하여 손가락 끝 점을 찾기 위해 아래와 같은 과정을 수행한다.

① 입력된 손가락 끝 점이 3번에 걸쳐 템플리트 매칭 시도
② 템플리트 매칭은 0~1 사이의 값을 가짐
③ 3번의 템플리트 매칭 시도 중 1에 가까운 값을 손가락 끝 점으로 인식

2. 손 영상에 대한 전처리

손 영상 집합의 정규화는 카메라로부터 연속적으로 얻어진 영상들을 모델 구성에 필요한 전처리 과정으로 시각 기반의 손 형상 인식에 필수적인 처리 과정이다. 특히 일반 환경에서 연속적인 영상들은 각각 서로 다른 많은 잡음들을 포함하기 때문에 정규화 처리에 의해 인식률이 크게 좌우될 수 있다. 정규화 과정은 크게 영상의 크기 정규화와 영상의 연산 속도를 줄이기 위한 색상정보 정규화 과정으로 나누어진다. 크기 정규화를 거친 손 영상들은 많은 잡음을 가지고 있기 때문에 잡음을 제거한 후 인식 속도를 개선하기 위하여 영상의 색상을 Gray Level로 정규화 한다. 또한 템프리트 매칭을 위한 모델영상의 정규화 과정으로 영상의 색상 정보를 2진화 시킨다. [그림 4-2]는 전처리 과정을 나타낸다.

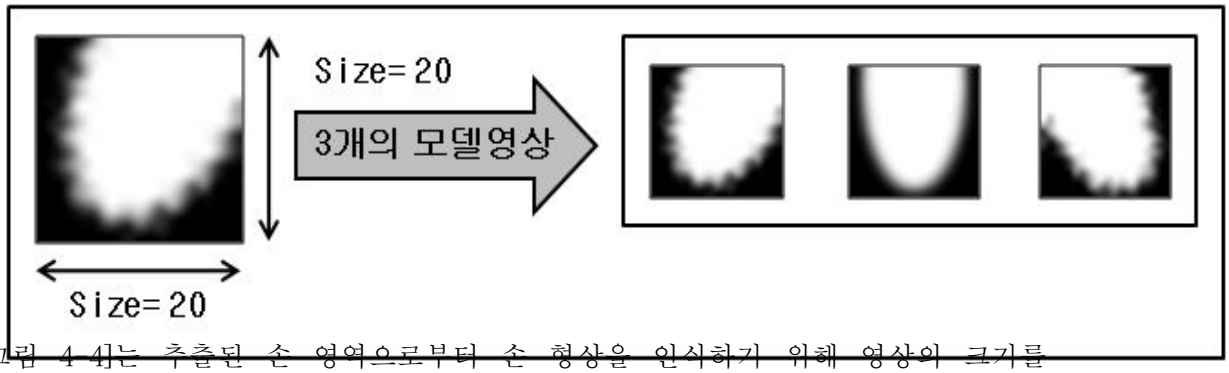


[그림 4-2] 템플리트 매칭과 손 형상 인식을 위한 전처리 과정

Collection @ chosun

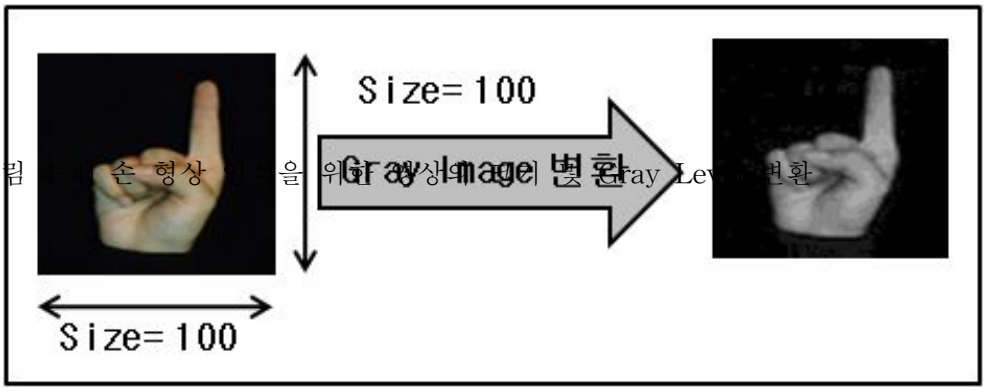
2-1. 손 영상의 크기 및 색상정보 정규화

본 논문에서는 1024*768의 입력영상으로부터 손 영역만 검출된 영상은 항상 일정치가 않기 때문에 영상을 일정한 크기로 정규화하고, 계산량을 줄이기 위해 영상의 크기를 100*100으로 정규화시켰다. [그림 4-3], [그림 4-4], [그림 4-5]는 영상의 크기와 색상정보의 정규화 및 손 형상 인식을 위한 모델 영상을 나타낸다. [그림 4-3]은 템플리트 매칭을 이용하여 손가락 끝 점을 인식하기 위해 3개의 모델 영상 크기를 20*20으로 정규화했고, 색상정보를 흑백영상으로 표현하기 위해 2진화 영상(Binary Image)으로 변환하였다.



[그림 4-3] 손가락 끝 점 인식을 위한 모델영상의 크기변환

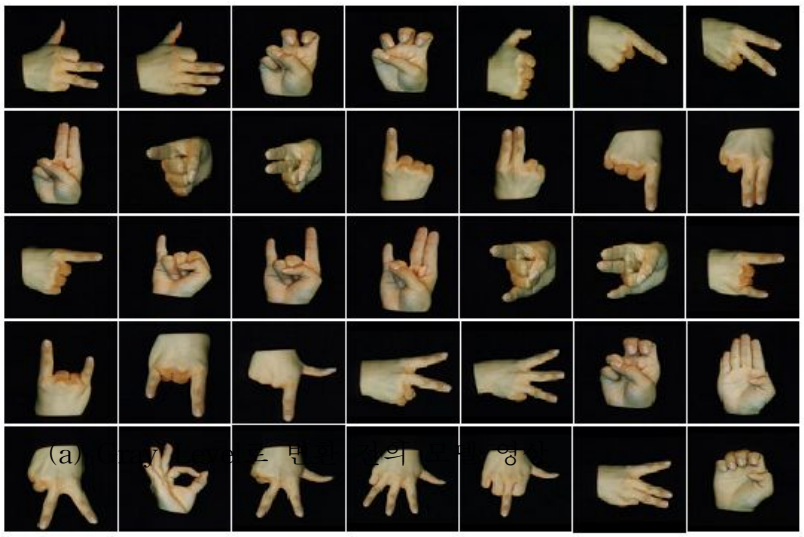
[그림 4-4]는 추출된 손 영역으로부터 손 형상을 인식하기 위해 영상의 크기를 100*100으로 정규화했고, 색상정보를 흑백영상으로 표현하기 위해 Gray Image로 변환 및 2진화하였다.



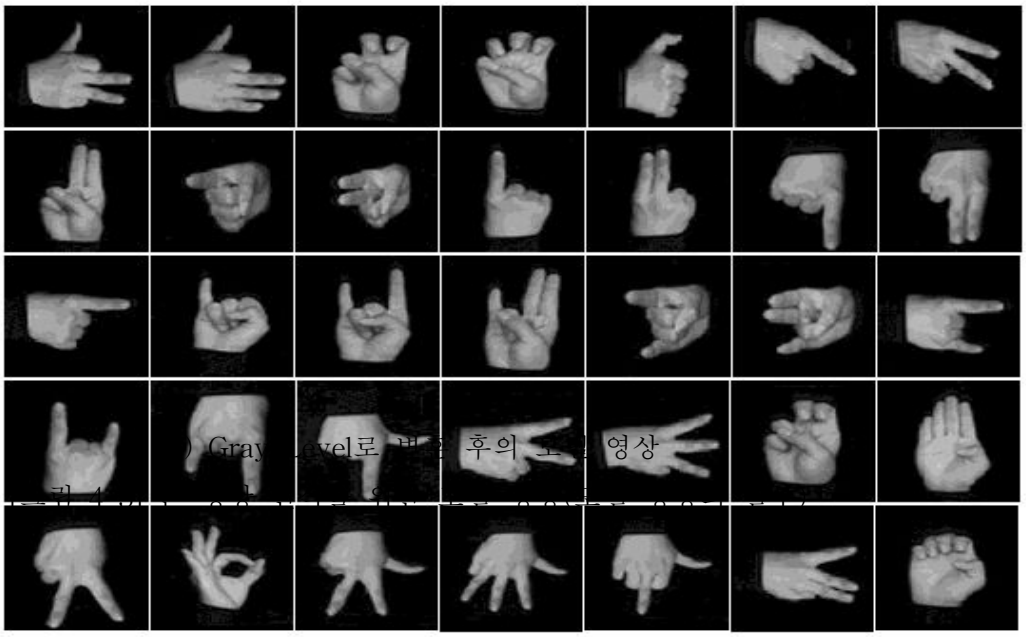
[그림 4-4] 손 형상 인식을 위한 영상의 크기 및 Gray Level 변환

Collection @ chosun

[그림 4-5]는 손 형상을 인식하기 위한 손 형상의 모델 영상이다. 모델 영상은 총 100개이지만 [그림 4-5]에서는 일부 영상만을 보여준다. [그림 4-5](a)는 Gray Image로 변환하기 전의 영상이며, (b)는 Gray Image로 변환한 영상이다.



(a) Gray Image로 변환 전의 모델 영상



(b) Gray Level로 변환 후의 모델 영상

[그림 4-5] 손 형상의 모델 영상

Collection @ chosun

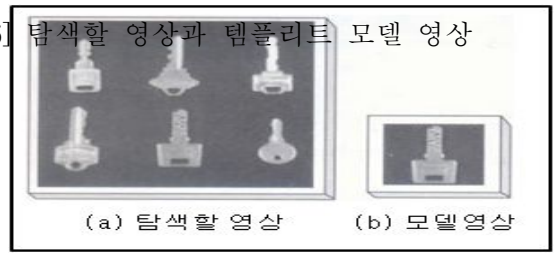
3. 템플리트 매칭(Template Matching)

3-1. 템플리트 매칭의 개요

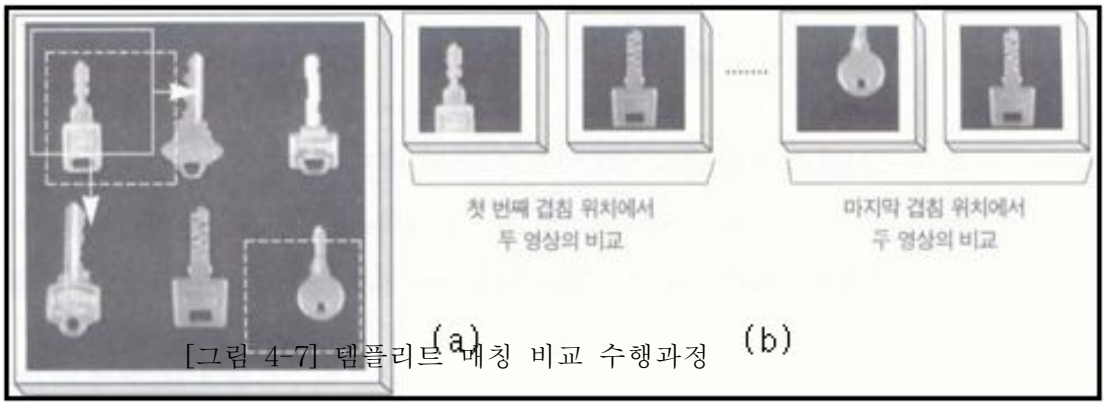
템플리트 매칭이란 검사할 영상이 주어졌을 때 미리 주어진 템플리트 영상(이미 알고 있는 영상패턴)을 이용하여 검사할 영상 내부에 있는 유사한 영상패턴을 찾는 것이며, 이때 템플리트는 일종의 모델(Model) 영상이다. 다음에 보여 줄 [그림 4-6]은 탐색할 영상과 템플리트 모델영상이다. [그림 4-6](a)는 한 장의 영상이 놓여져 있고 그 영상 안에서는 여러 가지 형태의 열쇠가 놓여져 있다. 이 영상에 대해서 내 사무실 열쇠가 영상 내에 어디에 위치하고 있는지 알고 싶다면, 먼저 알아야 할 것은 내 사무실 열쇠는 무엇이고 어떻게 생겼는가에 대한 것이다. 이것은 우리가 미리 알고 있어야 하는 지식(Knowledge)이다. 미리 알아야 하는 내 사무실 열쇠의 영상이 템플리트이다. 내 사무실 열쇠의 템플리트가 [그림 4-6](b)에 조그마한 영상으로 주어져 있다. 인간의 눈은 매우 정교하므로 아주 쉽게 템플리트와 유사한 패턴이 영상 내부에 어디에 위치하고 있는지 발견할 수 있다. 그러나 컴퓨터에게는 이것이 간단한 문제는 아니다.

템플리트 매칭이 어떻게 수행되어지는지의 과정은 [그림 4-7]에 나타냈다. 템플리트 영상을 검사할 영상에 겹쳐서 나타낸 것이 [그림 4-7](a)에 나타나 있다. 영상의 왼쪽 맨 위의 시작점에 조그마한 템플리트를 겹쳐놓고 검사할 영상의 겹쳐진 부분과 템플리트 영상을 서로 비교한다. 이러한 비교 기준치는 용도에 따라 적당하게 선택할 수 있다. 계산된 비교 기준치를 저장한 후에, 다시 템플리트를 한 픽셀 왼쪽으로 옮긴다. 그리고는 다시 검사할 영상의 겹쳐진 부분과 템플리트 영상을 비교한다. 이 과정을 전체 영역에 대한 비교가 끝날 때까지 반복 수행한다. [그림 4-7](b)는 맨 처음 시작 부분에서 겹쳐진 검사영상 부분과 오른쪽 아래의 맨 마지막의 부분에서 검사할 영상의 겹쳐진 부분을 나타내고 있다.

[그림 4-6] 탐색할 영상과 템플리트 모델 영상



Collection @ chosun



[그림 4-7] 템플리트 매칭 비교 수행과정

검사할 영상의 모든 영역에서 비교 기준치를 저장한 후, 이 값이 가장 최적인 위치에 우리가 찾고자 하는 모델 열쇠가 놓여 있다고 가정한다. 따라서 템플리트 매칭에서는 이 비교 기준치를 잘 선정하는 것이 중요해진다. 다음은 비교 기준치를 결정할 때 고려해야 하는 몇 가지 포인트를 나타낸다.

① 영상 잡음(Image Noise)에 둔감해야 한다.
② 밝기 변화(Intensity Variation)에 둔감해야 한다.
③ 계산량이 적어야 한다.

검사할 영상은 여러 가지 환경요인으로 영상 잡음(Image Noise)가 나타나게 되는데 이러한 노이즈에 영향을 받지 않는 기준치가 좋은 기준치이다. 또한, 조명의 변화로 인해 밝기의 변화가 검사영상에 나타날 때도 기준치에 이러한 영향이 적으면 좋다. 기준치를 계산할 때 과도한 계산량이 요구되면 수행과정에서 수행속도가 늦어지므로 고속 영상처리기를 만들기가 힘들어진다.

가. 비교 기준치

[그림 4-7](b)에 주어져 있는 검사할 영상의 겹쳐진 부분과 템플리트 영상을 서로 비교하려고 할 때 어떤 기준치를 사용해야 하는가에 대한 문제를 살펴보면 다음 두 가지가 가장 간단한 식이다(식 4-1)(식 4-2).

Collection @ chosun

$$MAD = \frac{1}{MN}\sum_{i=0}^{M}\sum_{j=0}^{N}|T(x_i,y_i) - I(x_j,y_j)| \quad \text{(식 4-1)}$$

$$MES = \frac{1}{MN}\sum_{i=0}^{M}\sum_{j=0}^{N}[T(x_i,y_i) - I(x_j,y_j)]^2 \quad \text{(식 4-2)}$$

템플리트 T와 검사할 영상의 겹쳐진 영상부분 I를 서로 겹쳐놓고 대응되는 픽셀의 밝기값을 서로 빼서 차이값을 더하는 것이다. M과 N은 템플리트의 가로 및 세로 크기이다. MAD(Mean Absolute Difference)는 차이의 절대값을 더하고 MSE(Mean Square Error)는 차이의 제곱값을 더하는 것이다. 당연히 템플리트와 겹쳐진 부분의 영상 밝기값이 커질 것이다. 이러한 값들의 계산을 통해 겹침이 발생했을 때 각각의 비교치를 계산할 수 있다.

나. 연산 횟수

위의 [그림 4-7]에서처럼 사무실 열쇠의 위치를 발견하기 위해서는 템플리트 검사할 영상 위에 모델영상을 겹쳐놓고 몇 번을 반복해서 비교해야 한다.

① 템플리트 크기 : $M * N$
② 검사할 영상의 크기 : $R * C$
③ 겹침이 발생하는 횟수 : $(R-M) * (C-N)$

예를 들어, $(M * N)$ 크기의 템플리트와 $(R * C)$ 크기의 검사할 영상이 주어져 있다고 가정하면, 서로 겹쳐서 비교해보아야 할 회수는 $(R-M) * (C-N)$번을 실행하여야 한다. 만일 템플리트의 크기가 100 * 100 픽셀이고 검사할 영상의 크기가 640 * 480 이라고 가정한다면 540 * 380의 겹침 횟수가 발생함으로 시간복잡도가 발생한다. 시간복잡도가 높으면 유사한 패턴을 발견하기 위한 계산시간이 오래 걸리게 된다.

앞에서 겹침 횟수가 $(R-M) * (C-N)$번 발생한다고 가정하였다. MAD인 경우, 각각의 겹침 상태에서 뺄셈을 $M * N$ 번을 만큼을 연산함으로 총 뺄셈회수는 (R

Collection @ chosun

$- M) * (C - N) * (M * N)$번의 뺄셈을 수행해야하므로 템플리트 매칭의 계산횟수가 많다는 것을 알 수 있다. 전산학의 시간복잡도에서 “빅오O”로 표현했을 때, O(n4)의 개념으로 “for″를 사용하는 반복문이 네 번 겹쳐서 사용되는 것과 같다. 따라서 보다 실용적인 템플리트 매칭 기술을 개발하기 위해서는 고속으로 수행되는 정합법이 필요하다.

```
① for(i=0; i<n; i++) sum = sum+i;  //O(n)
② for(i=0; i<n; i++) for(j=0; j<n; j++) sum = sum+i*j;  //O(n2)
③ for(i=0; i<m; i++) for(j=0; j<n; j++) sum = sum+i*j;  //O(mn)
④ for(i=0; i<m; i++) sum = sum+i;
                     for(j=0; j<n; j++) sum = sum+j;  //O(m+n)
```

다. 농담정규화 정합

앞에서 설명한 사무실 열쇠를 찾는 것[그림 4-7]과 비교하면서 설명하겠다. [그림 4-7]과 비교했을 때 [그림 4-8]은 전체적인 밝기가 밝아졌다는 것을 육안으로도 확인할 수 있다. 똑같은 검사영상을 밝은 곳에서 촬영한 영상이 [그림 4-8](a)이며, (b)는 템플리트 매칭의 모델영상이다.

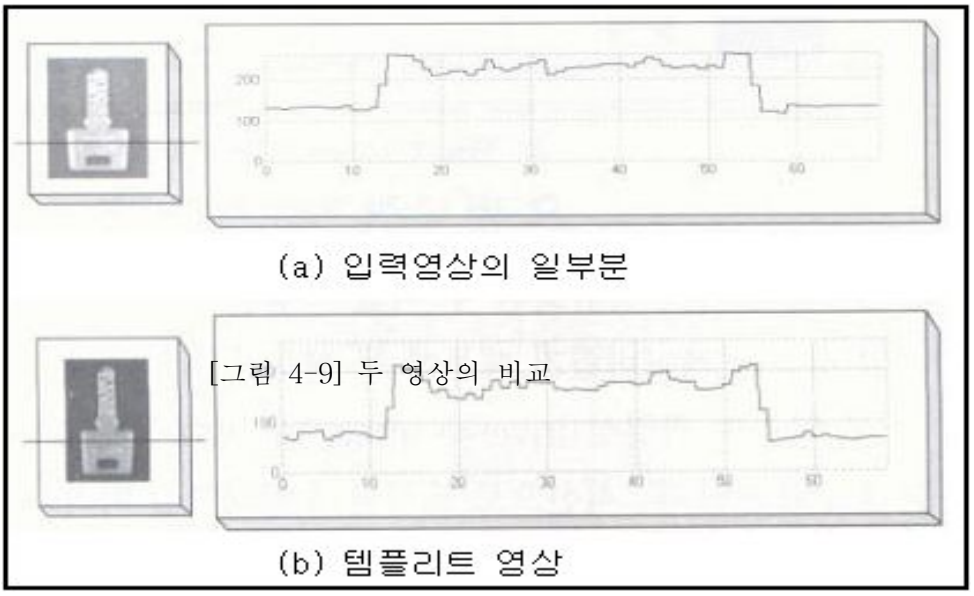
[그림 4-8] 밝은 곳에서 촬영된 검사영상

밝은 곳에서 촬영한 영상에서 MAD나 MSE를 사용하여 사무실 열쇠의 정확한 위치를 찾는다. [그림 4-8](c)그림에서 보여주는 사각형(Rectangle)은 MAD를 사용

Collection @ chosun

하여 사무실 열쇠를 발견한 위치를 보여주고 있으나, 잘못된 위치가 발견된 것을 확인할 수 있다. 감사할 영상의 밝기가 카메라로 찍을 때 더 밝은 조명 아래에서 찍혀졌기 때문에 전체적으로 밝아진 검사영상이 입력되었다. 실제로 [그림 4-8](a)는 형광등을 켜고 찍은 영상이다. 비교할 기준 템플리트 영상에 비교하여 전체적으로 밝기가 밝아졌기 때문에 더 이상 밝기의 차이값 만을 이용하는 기준치인 MAD나 MSE는 적용하기가 힘들다.

[그림 4-9]를 보면 템플리트 열쇠 영상과 입력영상에서의 모델 열쇠 영상이 놓여있는 부분만을 취해 두 영상의 밝기의 프로파일(Profile)을 비교한 그림을 보여준다. 영상위에 표시된 가는 수평 직선을 따라가면서 흑백 밝기의 값 크기를 그래프로 보여주고 있다. 입력영상에서 취한 부분은 밝기의 프로파일 값이 템플리트영상보다 일정한 값만큼 증가되어 있다는 것을 알 수 있고, 이러한 영상의 밝기 증가가 단순하게 두 영상의 밝기값의 차이만을 따지는 MAD나 MSE가 실패하도록 만드는 원인이 된다. 형광등을 켜고 다시 찍은 경우이기 때문에 검사할 영상의 밝기값이 영상의 모든 영역에서 전제적으로 증가되었다는 것을 알 수 있다. 따라서 입력영상에서 취한 비교할 영상 영역의 밝기 평균을 구하여 평균값을 빼버린 영상을 만들고 이 영상을 비교하기 위한 영상으로 사용하는 방법을 생각할 수 있다. 이때, 유사하게 템플리트 영상도 밝기의 평균을 계산하여 평균 밝기값을 템플리트 구성 화소에서 빼버린 정규화된 밝기 영상을 만드는 것이 가능하다.



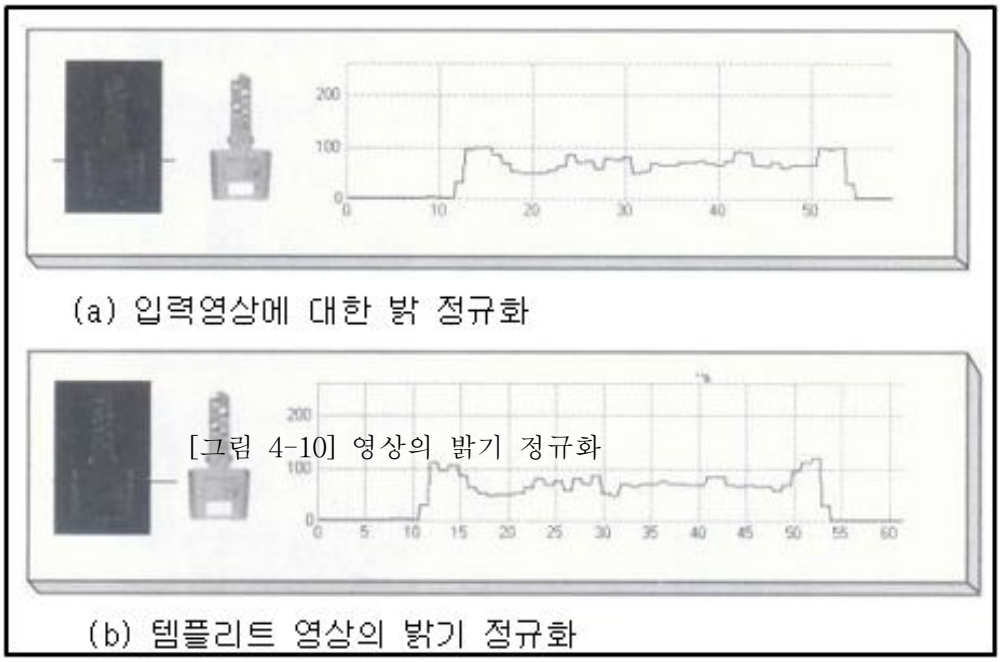
[그림 4-9] 두 영상의 비교

Collection @ chosun

이렇게 하여, 템플리트와 비교할 검사영상의 비교영역을 밝기에 대해 정규화시켜 비교한다면 전체적인 조명의 밝기가 높아지거나 낮아진다고 하여도 패턴의 비교가 가능할 것이다. 이러한 방법을 농담정규화 정합법(Normalized Gray-level Correlation: NGC)이라고 한다.

[그림 4-10]은 밝기가 정규화된 두 비교 영상을 보여주고 있다. [그림 4-9]의 두 영상에서 밝기의 평균을 제거했기 때문에 두 영상의 밝기는 어둡게 나타난다. 바로 옆에 알아보기 좋게 두 영상의 역영상(Inverse Image)을 보여주고 있다. 옆의 프로파일 그래프를 살펴보면 밝기가 보상된 후, 직선으로 표시된 수평라인을 따라가면서 밝기의 크기를 그래프로 그려보면, 두 영상의 프로파일은 거의 높이가 유사한 밝기 분포를 보여주고 있다. 따라서, 정규화된 밝기에 대해 다시 MAD나 MSE를 적용한다면 성공적인 정합이 발생할 수 있을 것이다.

영상처리를 2차원 신호처리(Two-Dimensional Signal Processing)이라고 부르는 사람도 많다. 영상신호도 신호라는 입장에서 보면 신호값들을 적당히 샘플링(Sampling)을 통해 취하고 취한 개수만큼의 다차원 공간을 정의하고 양자화된 크기값을 표시한다면, 하나의 영상은 이렇게 정의된 공간상의 하나의 점으로 표현할 수 있다.

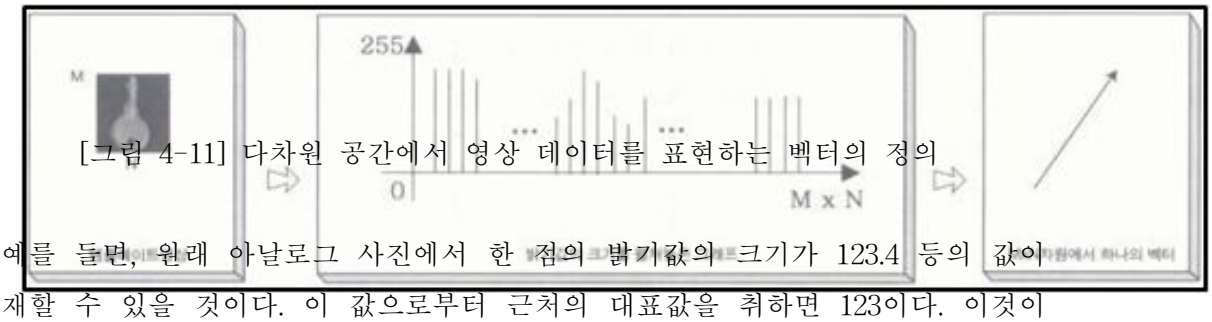


[그림 4-10] 영상의 밝기 정규화

Collection @ chosun

극단적으로 영상이 두 개의 화소값 만을 가지고 있다고 할 때, 2차원 평면상에 하나의 점으로 표시되는 벡터가 될 것이다. 영상 정합에서 사용되는 템플리트 영상은 M * N 크기의 화소수를 가진다. 화소의 밝기값이 각각의 대응되는 축에 대한 크기라면 템플리트 영상이란 M * N 크기의 차원을 가지는 다차원 공간상의 하나의 점이 되는 벡터이다. 따라서 위에서 나타낸 상관계수 r은 두 영상을 비교하는 영상 상관계수로도 사용 가능하다.

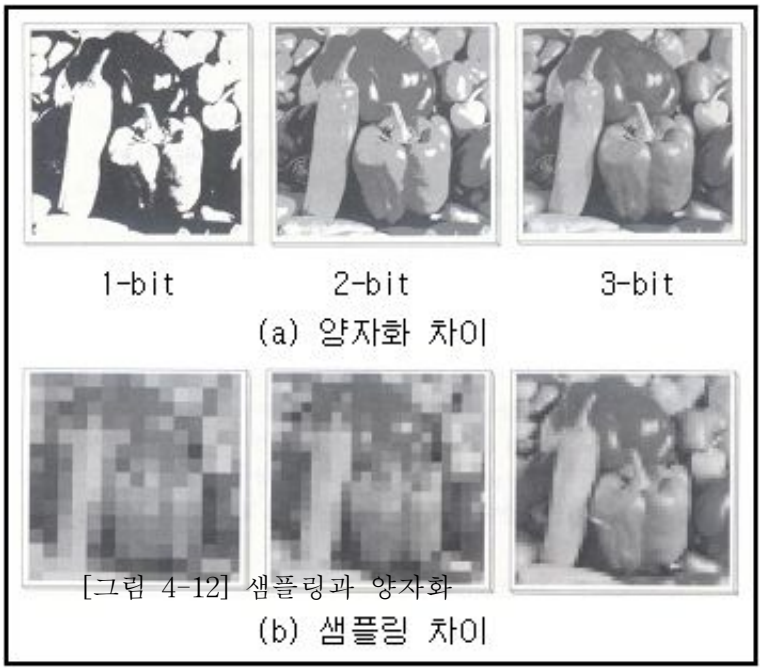
[그림 4-11]을 보면 열쇠를 표현하는 템플리트 영상이 하나주어져 있다. 이 영상의 크기는 M * N이다. 화소수가 전부 MN개가 있으므로 이 화소수를 수평축으로 하고 화소의 밝기값을 수직축으로 해서 펼친 그래프가 옆에 나타낸 밝기 그래프이다. 이때, 화소수 만큼 큰 이론적인 고차원을 하나 정의한다면 이 템플리트의 화소값 데이터는 MN-차원상에 놓여 있는 하나의 벡터가 된다. 따라서 템플리트 영상은 식에서 표현된 상관계수를 계산하기 위한 하나의 벡터 a가 된다. 단, 2차원에서 정의된 것이 아니라 MN-차원에서 정의된 벡터라는 점만 달라진다.



[그림 4-11] 다차원 공간에서 영상 데이터를 표현하는 벡터의 정의

예를 들면, 원래 아날로그 사진에서 한 점의 밝기값의 크기가 123.4 등의 값이 존재할 수 있을 것이다. 이 값으로부터 근처의 대표값을 취하면 123이다. 이것이 양자화이다. 2-bit 데이터를 가지고 123.4를 표현하면 0이나 255의 둘 중에 하나의 값이 될 것이다. 화소를 표현할 수 있는 데이터 비트(bit) 수를 크게 하면 밝기의 강도에 대한 정밀한 묘사가 가능하다. 또한, 사진을 표현하는 영상 화소수를 고정된 값이 아닌 다른 더 큰 수만큼 취할 수도 있을 것이다. 이러한 화소 해상도에 관련된 내용이 샘플링이다. 더 많이 취하면 데이터를 더 정밀하게 표현할 수도 있다. [그림 4-12]는 양자화와 샘플링의 차이에 따른 영상을 보여주고 있다.

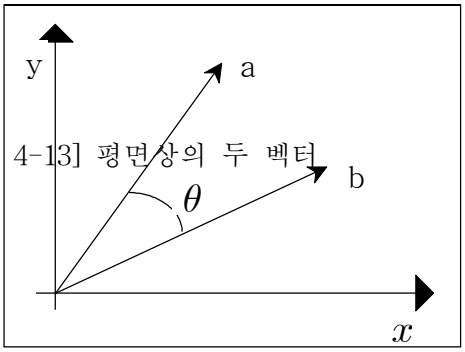
Collection @ chosun



[그림 4-12] 샘플링과 양자화

(1) 탬플리트 매칭의 수학적 배경

벡터는 크기와 방향을 가지고 있다. [그림 4-13]은 2차원 평면위에 놓여 있는 두 개의 벡터 성분 a와 b를 보여주고 있다. 이 두 벡터가 만일 동일한 벡터라면, 두 벡터의 크기와 방향은 서로 같을 것이다. 수학적으로 서로 다른 두 벡터의 비교는 벡터의 내적(Inner Product)을 이용한다. 수식은 (식 4-3)와 같이 나타낸다.



[그림 4-13] 평면상의 두 벡터

(식 4-3)

$$cos\theta = \frac{a \cdot b}{|a| \cdot |b|} = r$$

Collection @ chosun

의 값은 -1과 1사이의 값을 항상 가지므로 두 벡터의 비교 결과는 -1에서 1사이의 적당한 값이 계산되게 된다. 만일 두 벡터가 완전히 같은 크기와 방향을 가지고 있다면 각 는 가 되고, 의 값은 1이 될 것이고 완전히 같지 않다면 0이나 -1이 계산될 것이다. 따라서 위 두 벡터가 같은지 아닌지 판별하기 위한 기준으로 사용하는 것이 가능하다. 를 r이라 둔다면 이 r을 상관계수(Correlation Coefficient)라고 부른다. (식 4-3)의 와 는 두 벡터 의 크기를 나타낸다. 만일 a벡터가 축 값이 3이고 y축 값이 4라면 이 벡터의 크기는 5이고 방향은 의 값을 가질 것이다. 는 각각 방향의 단위 벡터이다.

$\cos\theta$ $\cos\theta$ 0° θ $\cos\theta$ $|a|$ $|b|$ a,b x

$$3/5i + 4/5j = 0.6i + 0.8j$$

i,j x,y

(2) 샘플링(Sampling)

연속적으로 변화하는 물리량 신호를 아날로그(Analog) 신호라고 한다. 아날로그 신호는 연속신호이기 때문에 디지털 컴퓨터에서 취급하기 위해서는 대표값들을 취하는 이산화(Dispersion)를 시켜주어야 한다. 이산화는 변수축에 대한 것과 값의 크기에 대한 것이 있는데, 변수축에 대한 것을 샘플링(Sampling)이라 하고 값의 크기에 대한 이산화를 양자화(Quantization)라고 한다. 이렇게 이산화된 신호를 디지털(Digital) 신호라고 한다. 영상신호에서 변수축은 시간축(Time-Domain)과 주파수축(Frequency Domain)이 주로 많이 사용된다. 값의 크기축은 보통 8-bit로 양자화하기 때문에 0~255 사이의 256개의 단계를 표현할 수 있다.

(3) 농담정규화 매칭법

템플리트와 입력영상에서 취한 비교되는 부분의 영상은 각각 밝기의 평균에 대해 보상(Compensation)되어 이 부분을 빼어버린 나머지 부분만을 가지고 비교한다. 농담정규화의 수식은 (식 4-4)과 같다.

(식 4-4)

여기서, 이다.

$$r = \frac{a \bullet b}{|a| \bullet |b|} = \frac{\sum_{i=0}^{M}\sum_{j=0}^{N} a(i,j)\,b(i,j)}{\sqrt{\sum_{i=0}^{M}\sum_{j=0}^{N} a(i,j)^2 \sum_{i=0}^{M}\sum_{j=0}^{N} b(i,j)^2}}$$

$$a(i,j) = g(i,j) - \overline{m},\ b(i,j) = t(i,j) - \overline{t}$$

Collection @ chosun

(식 4-5)에서 g와 t는 비교할 두 영상으로 입력영상 내부에서 취한 임의의 비교영역 부분과 템플리트 영상을 가리킨다. i, j는 수직 및 수평 방향으로의 인덱스로 픽셀좌표이다. $a(i, j)$는 입력영상에서 취한 비교부에서 $\bar{m}$을 빼버린 밝기값을 나타내며 $b(i, j)$는 템플리트 영상에서 평균밝기 $\bar{t}$를 뺀 밝기값을 나타낸다. 즉, 평균밝기를 빼서 정규화시킨 두 영상을 (식 4-4)의 두 벡터 비교식을 이용하여 유사도를 검사하는 수식이 (식 4-5)의 수식을 나타낸다. 템플리트의 크기는 $M * N$ 이다.

(식 4-5)

$$r = \frac{\left|(MN)\sum gt - \left(\sum g\right)\sum t\right|}{\sqrt{\left[(MN)\sum g^2 - \left(\sum g^2\right)\right]\left[(MN)\sum t^2 - \left(\sum t^2\right)\right]}}$$

(식 4-6)는 또 다른 정규화계수치 계산식으로 (식 4-5)보다 효과적으로 계산에 이용될 수 있다. 탐색 루프 안에서 계속 반복해서 계산이 필요한 부분은 $\sum g$, $\sum g^2$, $\sum gt$의 세 부분뿐이므로 이 부분을 제외한 나머지 부분들은 탐색 루프 밖에서 미리 계산해 놓는다면 농담정규화 정합의 계산량을 크게 줄이는 것이 가능하다. 실제적으로 탐색시 음수의 계수치는 관심의 대상이 아니므로 배제시킨다. 수식에 들어 있는 Square-root 연산은 계산량을 상당히 증가시키므로 이 값을 배제하기 위해 r 값을 제곱해 준다. 따라서 최종적인 정합의 계수치는 (식 4-6)와 같은 형태가 된다.

(식 4-6)

$$Score = \left[Max(r, o)\right]^2$$

3-2. 템플리트 매칭을 이용한 가상환경 제어

가상환경에서 방향전환을 하려면 키보드나 마우스를 이용하여 방향전환을 하지만 다소 번거로운 작업이므로 좀 더 편리한 작업환경을 제공하기 위하여 사용자의 손을 입력정보로 직접 사용한다. 정확한 방향전환을 하려면 손가락이 어느 방향을 지시하고 있는가를 정확히 인식해야 한다. 본 논문에서는 템플리트 매칭(Template Matching)을 이용하여 입력된 손 형상에서 손목의 중심과 손가락 끝 점(FingerTip)을 이용하여 원하는 방향을 가리킴으로써 그 방향을 알 수 있다.

Collection @ chosun

가. 가상환경 제어의 개요

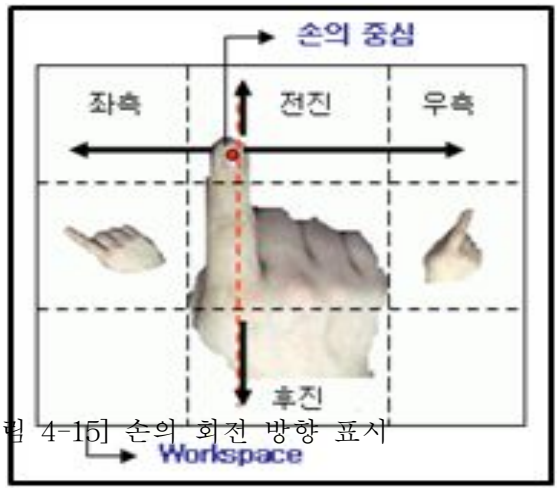
본 논문에서는 마우스, 키보드, 조이스틱 등의 조작 없이 사용자의 손을 직접 이용하여 전진, 후진 이동과 좌측, 우측 방향전환을 할 수 있는 시스템을 구현하고자 한다. 본 시스템의 구성은 [그림 4-14]와 같으며, 사용자가 책상에 앉아서 편리하게 사용할 수 있는 환경이다. 전면에 모니터가 있으며, 모니터 위에는 범용 USB 카메라가 달려 있고, 작업 공간은 카메라에 들어오는 영역 모두가 된다. 본 논문에서 제안하는 방법을 적용하기 위해 가상환경에서의 방향을 제어하는 시스템을 구현한다.

[그림 4-14] 방향 제어를 위한 시스템 구성

나. 가상환경 제어 정의

가상환경에서의 움직임은 초기화(대기상태), 전진, 후진, 좌측, 우측 5개의 명령으로 구성된다. 전진과 후진 이동은 초기상태의 손가락 끝 점의 위치를 기준으로 손가락 끝 점의 위치변화로 알 수 있으며, 좌측과 우측의 방향전환은 초기상태의 손가락 끝 점의 위치에서 좌측, 우측으로의 위치변화로 알 수 있다. 초기상태의 기준은 손목의 중심점과 손가락 끝 점의 중심을 직선으로 연결시킬 점이 초기상태의 기준값(0)이 된다. [그림 4-15]는 가상환경에서의 방향전환에 대한 명령 정의를 나타낸다.

Collection @ chosun



[그림 4-15] 손의 회전 방향 표시

다. 손가락을 이용한 방향 제어

가상환경 상에서 움직임 명령은 [표 4-1]과 같으며, 총 5개의 방향전환명령을 나타낸다. 전진과 후진은 초기영상의 손가락 끝점의 위치변화를 이용하여 결정한다. 즉, 초기영상의 손가락 끝점 보다 위로 움직이면 전진명령이며, 초기영상의 손가락 끝점 보다 아래로 움직이며 후진 명력으로 인식한다. 또한 좌측, 우측 방향전환은 기준점에 대해 손가락 끝 점의 상대위치를 이용하여 방향을 인식한다.

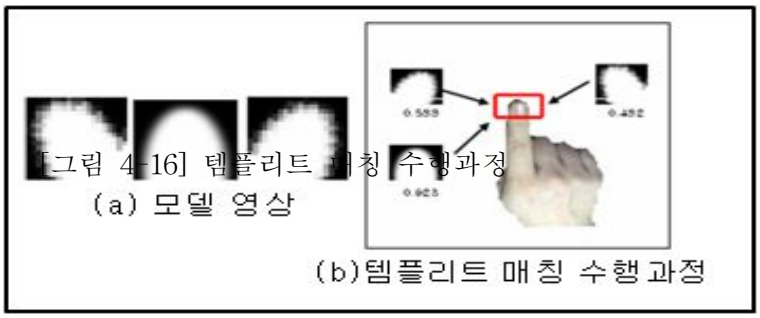
[표 4-1] 방향전환 명령어

명령어	Hand Pose	내용
대기상태		초기 입력 손 형상
앞으로 이동		손가락 끝 점이 앞으로 이동
뒤로 이동		손가락 끝 점이 뒤로 이동
좌측으로 이동		초기상태에서 손가락 끝 점이 좌측으로 이동
우측으로 이동		초기상태에서 손가락 끝 점이 우측으로 이동

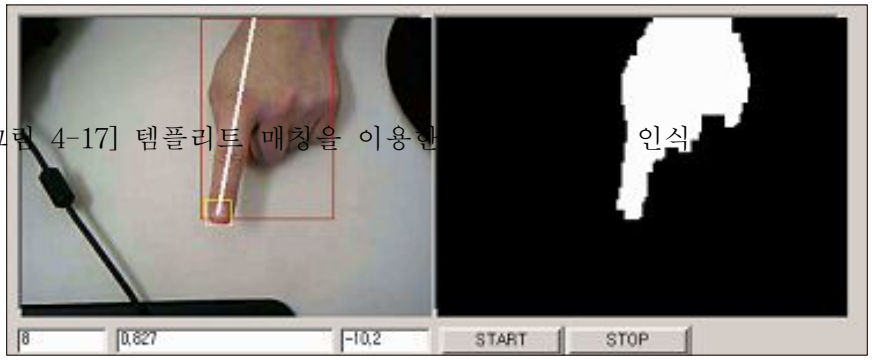
Collection @ chosun

라. 템플리트 매칭을 이용한 손가락 끝 점 탐색

본 논문에서는 손가락 끝 점을 찾기 위한 방법으로 템플리트 매칭(Template Matching)을 사용하였으며, [그림 4-16](a)와 같이 각기 다른 모양의 모델 3개를 사용하였다. 손 형상이 입력되면 손가락 끝 점에 대해서 각기 다른 3개의 모델이 손가락 끝 점 후보 영역들에 대하여 3번에 걸쳐 템플리트 매칭을 시도한다. 템플리트 매칭은 0~1 사이의 값을 가지도록 하였으며, 손가락 끝 점 인식은 3번의 템플리트 매칭 중에서 1에 가장 가까운 값을 갖는 모델이 우리가 찾는 손가락 끝 점의 영역이라고 가정했다. 단, 제안하는 방법에서 필요한 손가락 끝 점의 개수는 하나이기 때문에 손가락 끝 점은 단 하나의 영역만을 갖는다고 가정한다. [그림 4-16]은 3개의 모델에 대해 템플리트 매칭이 수행되는 과정을 나타낸다. (a)는 템플리트 매칭을 위해 각기 다른 모양을 갖는 3개의 모델 영상이며, (b)는 3개의 모델 영상이 입력된 영상과 템플리트 매칭을 시도하는 과정을 나타낸다. [그림 4-17]은 [그림 4-16]의 템플리트 매칭 수행과정에 의해 탐색되어진 손가락 끝 점 탐색결과를 실험 영상으로 보여주고 있다.



[그림 4-16] 템플리트 매칭 수행과정



[그림 4-17] 템플리트 매칭을 이용한 인식

Collection @ chosun

마. 손가락 끝 점을 이용한 방향 탐색

본 논문에서는 손가락이 가리키는 방향을 계산하기 위해 손목의 중심점과 손가락 끝 점을 이용하였다. 손목의 중심점과 손가락 끝 점까지의 거리를 진선으로 연결한 후 그 진선이 이루는 방향을 손의 방향으로 가정하였다. (식 4-7)과 같이 distance(거리)는 손목의 중심점과 손가락 끝 점의 중심점 사이의 거리를 나타냈으며, h_x 는 손목 중심의 *x*좌표, h_y 는 손목 중심의 *y*좌표, f_x 는 손가락 끝 점의 *x*좌표, f_y 는 손가락 끝 점의 *y*좌표이다.

손의 방향은 초기값(0)에서 손가락을 좌측(-) 또는 우측(+)으로 움직였을 때 그 각의 변화량을 이용한다. 변화량은 $\Delta\theta, -\Delta\theta$ 을 이용하여 손목 중심의 y 좌표인 h_y 와 손가락 끝 점의 y 좌표인 f_y 의 차를 *distance*로 나눔으로써 초기값에서 각의 변화율($a\cos$) 만큼을 계산하여 구할 수 있다. (식 4-8)을 이용하여 각의 초기값(0)에서 각의 변화율을 구하였다.

$$\Delta\theta, -\Delta\theta$$

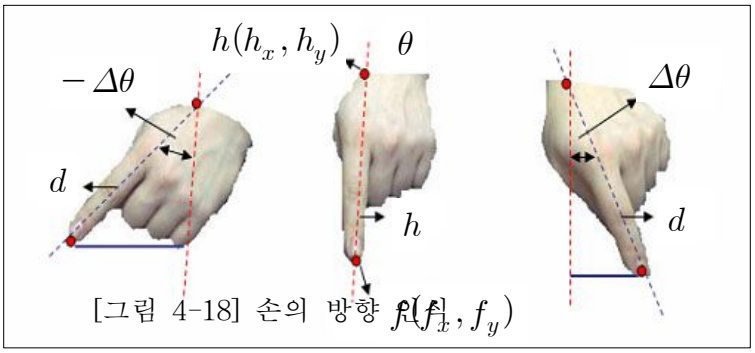
(식 4-7)

$$distance = \sqrt{(h_x - f_x)^2 + (h_y - f_y)^2} \quad \text{(식 4-8)}$$

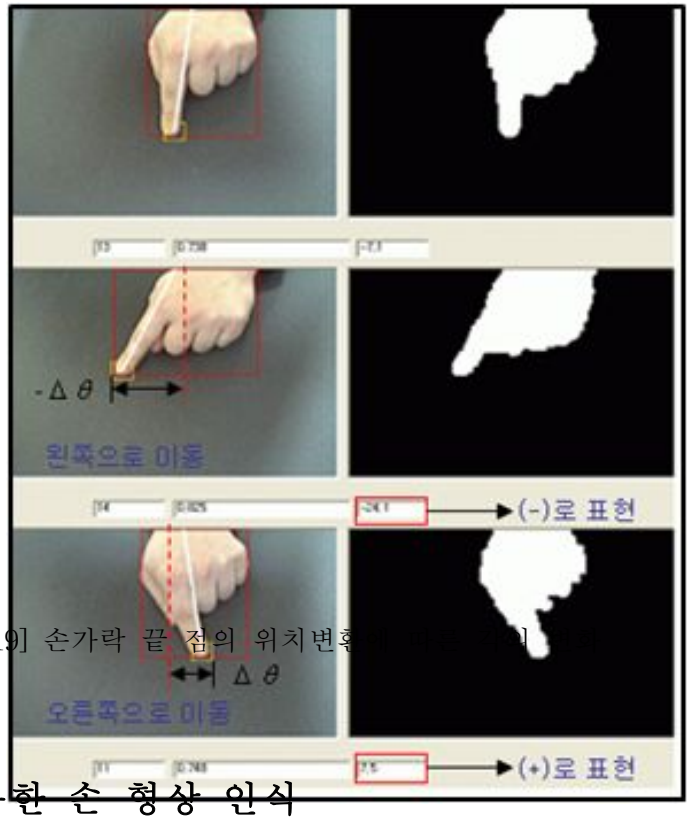
[그림 4-18]은 초기값(0)에서 각의 변화가 생겼을 때 그 각의 변화율($angle = a\cos\left(\frac{h_y - f_y}{distance}\right)$)을 구하기 위한 과정을 보이고 있으며, [그림 4-19]는 시스템상에서 방향의 변화에 따른 초기값(0)의 변화를 보여주고 있다. 오른쪽으로의 변화는 (+)값으로 표현되고, 왼쪽으로의 변화는 (-)값으로 표현되고 있음을 보여주고 있다.

$$\Delta\theta, -\Delta\theta$$

Collection @ chosun



[그림 4-18] 손의 방향 $f(f_x, f_y)$



[그림 4-19] 손가락 끝 점의 위치변화에 따른 각도 표현

4. 신경망을 이용한 손 형상 인식

4-1. 제안한 신경망 알고리즘

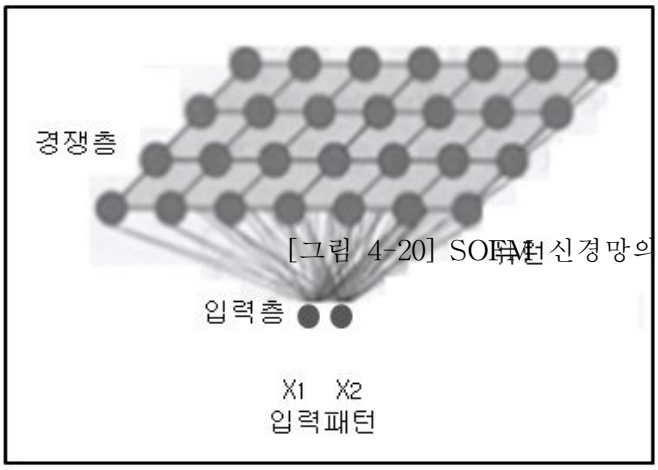
본 논문에서는 손 형상을 정확히 분석하고 인식하기 위해서 자기 조직화 특징

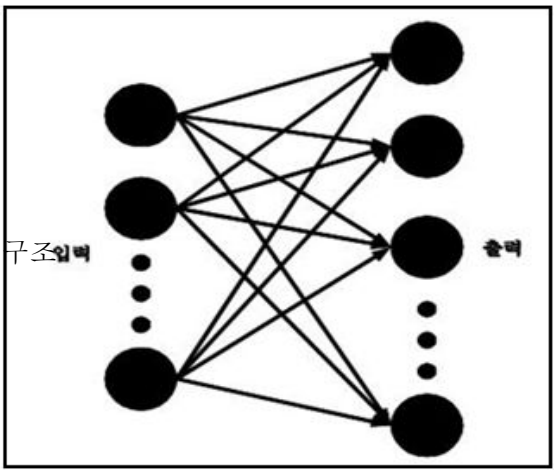
Collection @ chosun

지도(SOFM : Self Organized Feature Map)라는 신경망 알고리즘을 사용한다[43].

신경망의 학습방법에는 크게 입력패턴과 함께 목표패턴을 주는 교사학습(Supervised Learning)이 있다. 교사학습법은 출력결과가 의도하는 결과와 일치하는 지를 지속적으로 검사하여 일일이 학습에 관여하여 지나치게 데이터의 선정시간이 길어지는 단점이 있다. 이에 Kohonen은 입력패턴을 표현할 수 있는 특성지도(Feature Map)를 만들기 위해서 자기 스스로 조직화하는 신경 분류회로를 만들었다. 그것이 바로 자기 조직화 특징 지도이다.

SOFM은 인간의 뇌와 같은 맵의 형태로 구성되며, 입력 뉴런들, 출력 뉴런들, 그리고 입·출력 뉴런 간에 연결된 시냅스들로 구성되어 있다. SOFM 알고리즘의 학습 단계에서 다차원 공간상의 특징 벡터로 표현되는 객체들이 경쟁층에서 스스로 유사한 패턴끼리 2차원의 특징 지도를 조직화하여 영역지도를 형성한다. 그리고 인식단계에서는 이미 학습 단계에서 훈련된 연결 가중치 하에서 미지의 특징 벡터에 대하여 경쟁층에서 반응이 일어나는 위치를 통하여 해당 클래스를 인식한다. SOFM 신경망은 오류 역전파(Error Backpropagation) 알고리즘에 의하여 학습되어지는 MLP(Multi-Layer Perceptron)와는 달리 계층적인 구조를 가지고 있지 않다. SOFM 신경망의 구조는 [그림 4-20]과 같다. 첫 번째 층은 입력층이라고 하며, 두 번째 층은 2차원의 격자로 되어있는 경쟁층으로 되어 있으며, 모든 연결들은 첫 번째 층과 두 번째 층의 방향으로 완전 연결(Fully Connected)되어 있는 구조를 가진다. 더 높은 차원의 격자들을 가지는 구조도 구현이 가능하지만 3차원 이상의 격자 구조는 현재 널리 사용되고 있지 않다.





[그림 4-20] SOFM 신경망의 구조

Collection @ chosun

[그림 4-20]은 대표적은 형태의 SOFM을 나타내며, X_i는 입력 벡터, W_{ik}는 시냅스의 Weight를 나타낸다. 입력 뉴런의 수는 입력 변수의 차원과 동일하고, 출력 뉴런의 수는 입력 데이터를 대표하는 데이터 패턴의 수와 같다. 그리고 입력 뉴런과 출력 뉴런은 시냅스로 연결되어 있다. 이와 같은 형태로 구성된 신경망 모델은 출력 뉴런들이 승리하기 위하여 서로 경쟁을 하는 경쟁학습(Competitive Learning)을 하게 된다. 입력패턴이 주어졌을 때 출력 뉴런들은 경쟁을 한 후 단 한 개의 출력 뉴런만이 승리하게 된다. 그러나 여러 경쟁학습 알고리즘에 나타나는 특정 출력 뉴런들만 반복적으로 학습을 하게 되는 현상이 나타나는데 이러한 현상을 줄이기 위하여 승리 뉴런뿐만 아니라 이웃하는 뉴런도 함께 학습을 시켜줌으로써 출력 뉴런의 편중학습과 미상용 문제를 해결한다.

입력 영상의 특징 벡터와 모델 영상의 특징 벡터 사이의 거리를 비교하기 위하여 (식 4-9)를 이용하여 전체 모델 영상에 대한 평균 영상 를 구성한다. [그림 4-21]은 전체 모델 영상에 대한 평균 영상이며, 모델 영상은 일부만 나타낸다.

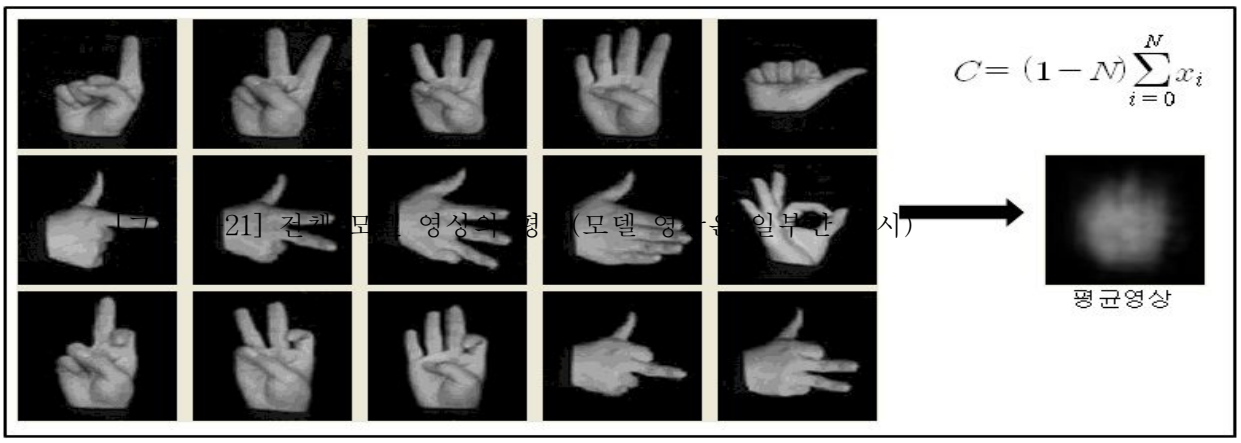
C

(식 4-9)

$$C = (1-N)\sum_{i=0}^{N} x_i \qquad \text{(식 4-10)}$$

여기서 N은 손 영상의 개수이며 x_i는 입력 영상이다.

$$X^{\triangle} = \left\{x_{1,1}^{(1)} - c, x_{2,1}^{(1)} - c, \ldots x_{R,1}^{(1)} - c, \ldots x_{R,L}^{(p)} - c\right\}$$



[그림 4-21] 전체 모델 영상의 평균 (모델 영상은 일부만 나타냄)

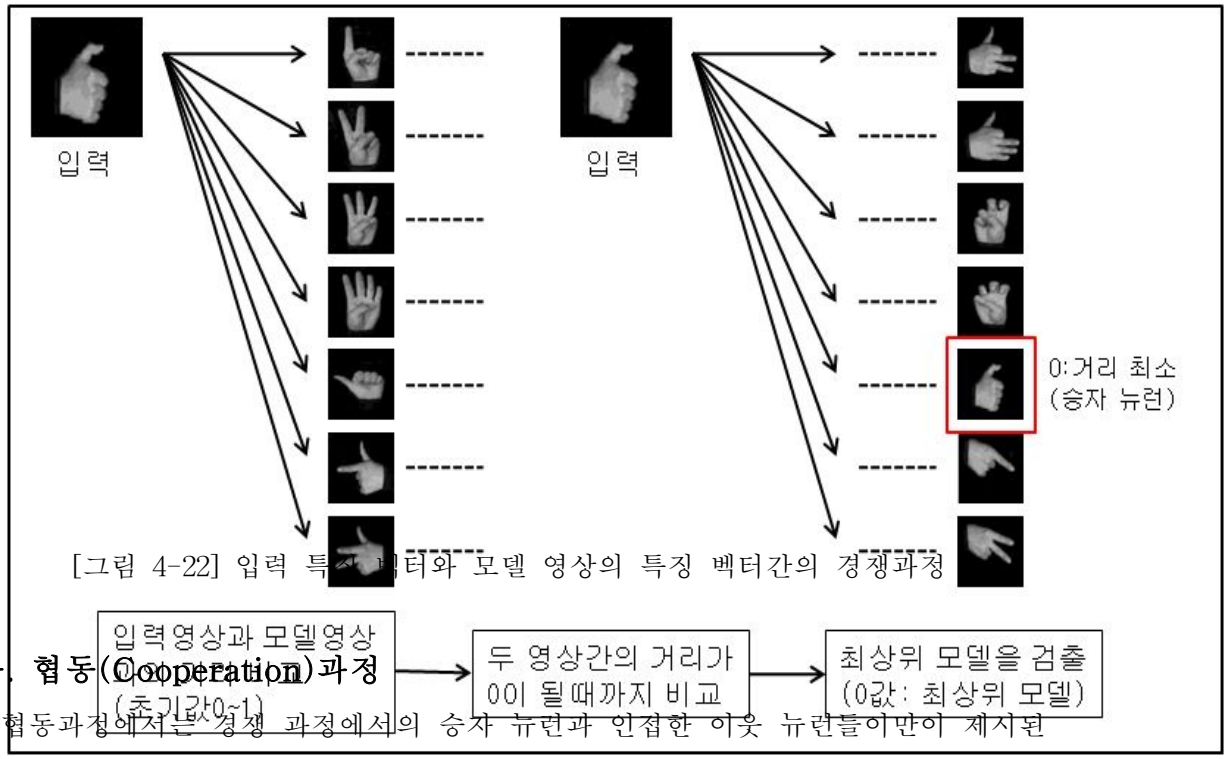
Collection @ chosun

4-2. SOFM의 학습 과정과 장·단점

SOFM의 학습 과정은 경쟁 과정, 협동 과정, 적응 과정의 세 가지 과정으로 이루어진다.

가. 경쟁(Competition)과정

SOFM에서 경쟁층 상에 있는 각 뉴런은 입력 공간을 구성하는 N차원의 특징 벡터와 연결 강도에 의하여 연결된다. 연결 강도는 0에서 1사이(0~1)로 정규화된 임의의 값으로 초기화되어 있다. 학습 단계에서 어떤 입력 패턴이 주어지면 모든 뉴런의 연결 강도와의 거리를 계산하여 거리가 최소가 되는 뉴런이 승자가 되는 경쟁의 규칙을 가진다. [그림 4-22]는 입력 특징 벡터와 모델 영상의 특징 벡터 사이의 거리를 비교하여 최상의 모델인 승자 뉴런을 찾는 경쟁과정을 보여준다.



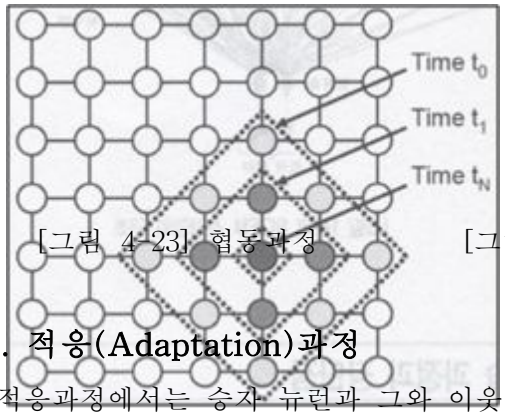
[그림 4-22] 입력 특징 벡터와 모델 영상의 특징 벡터간의 경쟁과정

나. 협동(Cooperation)과정

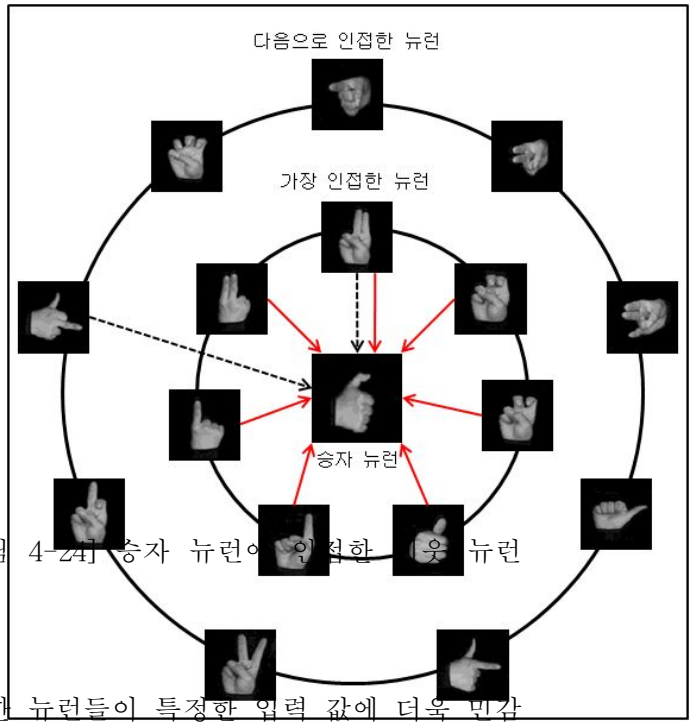
협동과정에서는 경쟁 과정에서의 승자 뉴런과 인접한 이웃 뉴런들이만이 제지된

Collection @ chosun

입력 벡터에 대하여 학습이 허용된다. 즉, 유사한 특징 패턴에 민감하게 경쟁층에서 지도를 형성하기 위하여 승자 뉴런은 이웃한 뉴런을 정해진 함수에 의하여 결정하고 그 뉴런의 연결 강도를 갱신한다. Kohonen의 학습은 승자 독점(Winner Take All)이므로 승자만이 출력을 낼 수 있으며, 승자와 그의 이웃들만이 연결 강도를 조정할 수 있도록 한다. [그림 4-23]은 SOFM 신경망의 협동과정을 나타내며, [그림 4-24]는 승자 뉴런에 가장 인접한 이웃 뉴런을 나타낸 것으로 점선은 승자 뉴런과의 거리를 의미한다. 점선이 짧을수록 승자 뉴런과 인접한 뉴런이다.



[그림 4-23] 협동과정

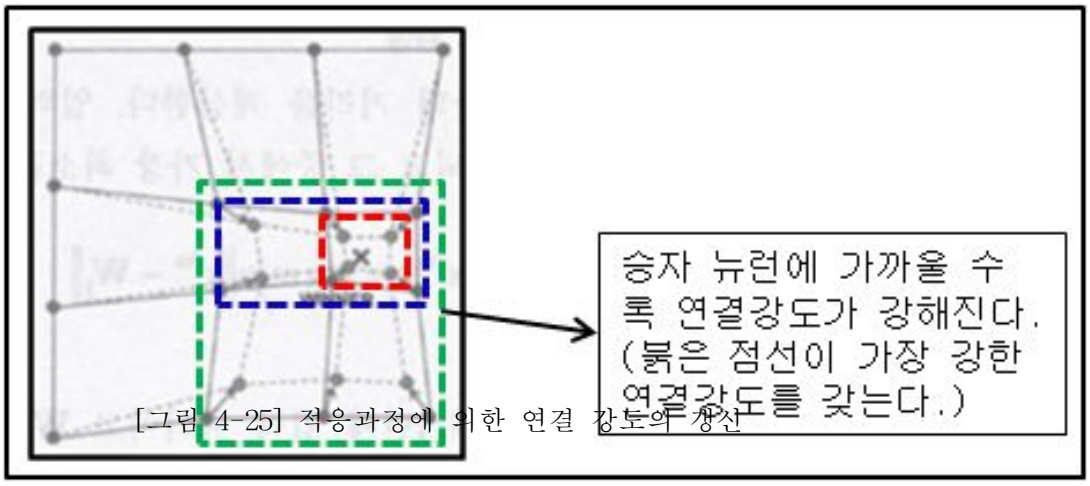


[그림 4-24] 승자 뉴런에 인접한 이웃 뉴런

다. 적응(Adaptation)과정

적응과정에서는 승자 뉴런과 그와 이웃한 뉴런들이 특정한 입력 값에 더욱 민감해지도록 활성 함수를 적응시켜 연결 강도를 갱신한다. 이 과정을 통하여 승자 뉴런과 인접해 있는 뉴런들은 멀리 떨어져 있는 뉴런보다 더 강하게 적응이 된다. 적응의 크기는 학습률로 제어되며 SOFM의 수렴 속도를 줄여주는 역할을 한다. [그림 4-25]의 격자는 모두 연결 강도를 의미하며, [그림 4-24]에서 실선 화살표는 가장 인접한 이웃 뉴런들이 멀리 있는 이웃 뉴런들보다 연결강도가 세다는 것을 나타낸다.

Collection @ chosun



[그림 4-25] 적응과정에 의한 연결 강도의 갱신

라. SOFM의 장・단점

첫 번째, SOFM 신경망은 역전파(BackPropagation) 모델과는 달리 여러 단계의 피드백이 아닌 단 하나의 전방 패스(FeedForward Flow)를 사용하는 구조를 이루고 인식 수행이 상당히 빠름으로 실시간 학습 처리를 할 수 있는 모델이다. 두 번째, SOFM 신경망은 연속적인 학습이 가능함으로 입력 데이터의 통계적 분포가 시간에 따라 변하면 자동적으로 이러한 변화에 적응하게 된다. 그러나 SOFM은 학습 단계에서 많은 시간을 필요로 하는 등 실제 문제에 적용하기는 쉽지 않다. 하지만 인간의 두뇌 작용을 고찰하기엔 매우 유용한 모델이다.

4-3. SOFM의 학습 단계

SOFM의 학습 단계는 아래와 같이 3단계로 이루어져 있다. [표 4-2]는 SOFM 학습 단계를 간략하게 나타낸 것이다.

[표 4-2] SOFM의 학습 3단계

1단계	연결 강도 초기화
2단계	입력 벡터가 제시되고 경쟁
3단계	선택된 뉴런과 그 이웃 반경 내의 뉴런들의 연결 강도를 갱신(협동과 적응 과정)

Collection @ chosun

1단계: 연결 강도 초기화

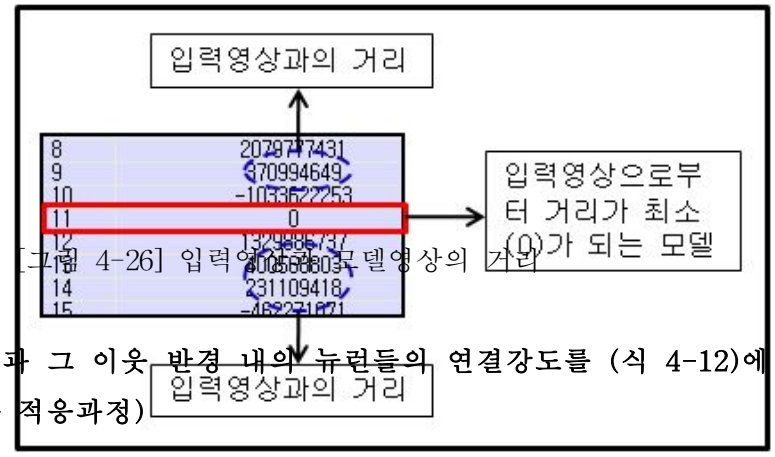
N개의 입력으로부터 M개의 출력 뉴런 사이의 연결강도를 작은 값을 임의 수로 초기화한다. 초기의 이웃 반경은 모두 뉴런들이 포함될 수 있도록 충분히 크게 잡았다가 함수 $h_{ik}(t)$에 따라 점차 줄어든다.

2단계: 입력 벡터가 제시되고 경쟁

입력 벡터와 모든 뉴런들간의 거리를 계산한다. 입력과 출력 뉴런 K사이의 거리는 (식 4-11)와 같이 계산되고 그 중에서 가장 최소값이 선택된다. [그림 4-26]은 [그림 4-22]의 입력 영상과 모델 영상사이의 거리를 계산하여 가장 최소값(0)이 되는 모델(최상위 모델)을 찾은 것이다.

(식 4-11)

여기서 arg는 $i(X^{(\in)})$ 성이 간의 연결강도 벡터이고 $\arg\min_j \|X^{(\in)} - W_j\|$ 시간 t 에서의 i 번째 입력 벡터이고 W_j 는 시간 t 에서의 i 번째 입력 벡터와 j 번째 출력 뉴런 사이의 연결 강도이다. 모든 연결 강도와의 거리를 계산하여 최소 거리에 있는 출력 뉴런을 선택한다. $X^{(\in)}$



[그림 4-26] 입력영상과 모델영상의 거리

3단계: 선택된 뉴런과 그 이웃 반경 내의 뉴런들의 연결강도를 (식 4-12)에 의해 갱신(협동과정과 적응과정)

Collection @ chosun

$$W_k = W_k + \eta(t) \cdot h_{ik}(t) \cdot \left(X^{(\in} - W_j\right) \quad \text{(식 4-12)}$$

$(t) \cdot j_{ik}(t)$는 0과 1사이의 값을 가지는 이득항(Gain Term)인데 시간이 경과함에 따라 점차 작아진다.

또한, SOFM에서는 Weight가 위상적인 배열을 형성하는 배열 단계와 정교한 맵을 구성하는 수렴 단계가 있는데, 일반적으로 전자의 경우가 입력 벡터 패턴에 대하여 적용하기 위한 초기 단계이므로 Weight를 크게 변화시키기 위하여 큰 이득항 값을 가지는 반면 수렴 단계에서는 배열 단계보다 학습 반복 회수가 상대적으로 길어야 한다. 현재 널리 사용되고 있는 학습 이득항에는 지수 함수적으로 학습 이득을 감소시키는 함수(학습률 감소 함수)를 사용한다(식 4-13).

(식 4-13)

$$\eta(t) = \eta_0 \exp\left(-\frac{t}{T_1}\right)$$

$\eta(t)$:학습률, η_0:초기 학습률, t:학습 반복 회수, T_1:학습률의 사상수

출력 뉴런의 편중학습과 미사용 문제를 보완하기 위하여 승리 뉴런과 함께 이웃 뉴런을 학습하게 된다. 이때 학습의 범위인 $\sigma(t)$도 역시 학습이 진행됨에 따라 학습 이득항과 같이 감소시키는 함수(이웃 감소 함수)를 사용한다(식 4-14).

(식 4-14)

$$\sigma(t) = \sigma_0 \exp\left(-\frac{t}{T_2}\right)$$

σ_0:초기 학습 이웃의 범위, t:학습 반복 회수, T_2:이웃 뉴런 학습의 사상수

일반적인 이웃 뉴런의 개수(σ_0)는 배열 단계에서는 출력 뉴런 수의 반 이상을 선택하는 것이 적정하며, 수렴 단계에서는 승리 노드 혹은 주위에 있는 1개의 노드만 포함하게 된다. 이 두 개의 설정값을 정한 후에는 학습 알고리즘을 이용하여 학습

Collection @ chosun

하게 되는데 SOFM을 이용해 학습하는 동안 승리 뉴런을 기준으로 주어진 이웃 내의 출력 뉴런에 해당하는 시냅스에 대하여 학습되는 이득률을 어떻게 결정할 것인지는 정교한 자기 상관지도를 만들기 위해서 필요한 결정요소가 된다.

학습되는 양을 결정하는 방법에는 첫 번째로 시냅스 Weight를 학습하는 방법 중 가장 간단한 방법인 거리와 무관하게 동일한 이득항으로 갱신하는 방법이다. 두 번째는 학습시 시냅스의 Weight를 학습할 때 승리 뉴런으로부터 떨어진 거리에 따라 학습되는 이득을 다르게 적용하는 방법이다. 거리에 따라 다른 이득으로 학습되는 방법을 적용하였을 경우에는 학습 알고리즘에서 시냅스 Weight 학습 함수는 (식 4-15)과 같이 나타낼 수 있다.

(식 4-15)

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t)\, m_{ij}(x_i(t) - w_{ij}(t))$$

m_{ij} :이웃 뉴런의 이득

SOFM을 실행모드에서 동작시킬 경우 시냅스 Weight를 변화시키지 않으며, 시냅스 Weight들과 입력 벡터를 비교하여 입력 벡터와 가장 근접한 시냅스 Weight를 가지는 승리 노드를 선택하는 기능만을 수행한다. [표 4-3]은 SOFM의 알고리즘이며, [그림 4-27]은 SOFM 알고리즘의 순서도이다.

[표 4-3] SOFM 알고리즘

1. i개의 입력 뉴런과 j개의 출력 뉴런 사이의 연결강도를 임의의 작은 값으로 초기화 한다.
2. 새로운 학습벡터를 입력한다.
3. 학습벡터와 각 출력 Weight 간의 유클리디언 거리를 구한다.

$$d_j = \sum_{i=0}^{N-1} \left(X_i(t) - w_{ij}(t)\right)^2$$

$X_i(t)$: 시각 t에서의 입력

$w_{ij}(t)$: 시각 t에서의 입출력 사이의 연결강도

4. 최소거리를 갖는 출력 뉴런을 선택한다.

Collection @ chosun

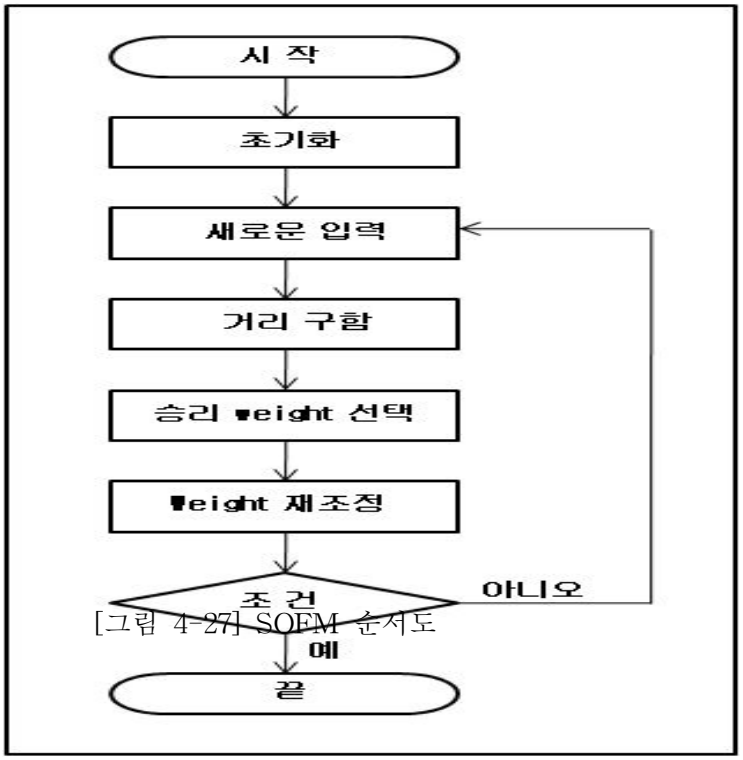
5. Weight를 재조정한다.

$$w_{ij}(t+1) = w_{ij}(t) + a(t)m(t)\left(x_i(t) - w_{ij}(t)\right)$$

이때, $a(t)$: 학습률

$m(t)$: 이웃

6. 2로 가서 조건이 만족될 때까지 반복한다.



[그림 4-27] SOFM 순서도

Collection @ chosun

V. 실험결과

실험은 범용 USB 카메라(초당 30 프레임)를 장착한 펜티엄IV 2.4GHz 메모리 1GB를 가진 PC에서 수행하였으며, 프로그램은 Visual C++를 사용하여 구현되었다. 실험에 이용된 영상은 첫째, 템플리트 매칭을 이용하기 위해 [그림 4-3]과 같이 모델 영상의 크기를 20*20으로 정규화했으며, 흑백영상의 모델 영상을 만들기 위해 모델 영상에 대해 2진화(Binary Image)를 하였다. 둘째, 손 형상을 정확하게 인식하기 위해 [그림 4-4]와 같이 모델 영상의 크기를 100*100으로 정규화했으며, 인식 속도를 높이기 위해 영상을 Gray Level로 변환하였다. 손 형상 인식에 사용되는 모델 영상은 총 100개의 영상으로 이루어져 있으나 본 논문에선 일부의 모델 영상만을 담았다. 카메라로 들어오는 영상을 입력정보로 사용하였으며, 피부색 정보를 이용하여 영상의 전체 영역에서 손 영역만을 분리하였다. 분리된 손 영역에 대해 2진화 영상을 만든 후 템플리트 매칭을 위한 모델 영상으로 매칭을 시도하여 손가락 끝 점을 찾아 가상환경을 제어한다. 또한, 분리된 손 영역으로부터 손 형상을 정확하고 안정적으로 인식하기 위해 SOFM(Self Organized Feature Map)라는 신경망 알고리즘을 사용하였으며, SOFM 신경망 알고리즘을 사용함으로써 입력영상에 대한 인식률을 향상시켰다.

1. 템플리트 매칭에 의한 가상환경 제어

본 논문에서는 템플리트 매칭을 이용하기 위해 [그림 4-3]과 같이 모델 영상의 크기를 20*20으로 정규화했으며, 피부색 정보로 손 영역을 추출한 후 템플리트매칭을 이용하여 손가락 끝 점을 찾은 후 손가락 끝을 이용하여 전진, 후진, 좌측, 우측 방향을 탐지한다. 방향 선택에 있어서 마우스나 키보드 대신 직접 사용자의 손으로 방향을 선택함으로써 사용자에게 좀 더 편리한 환경과 효율적인 인터페이스를 제공한다. 본 시스템은 가상환경 상에서 손을 이용하여 방향을 지시하는 Navigation 기능, 3D Game, 간단한 Window System 등을 제어하는 곳에 다양하게 응용될 수 있다.

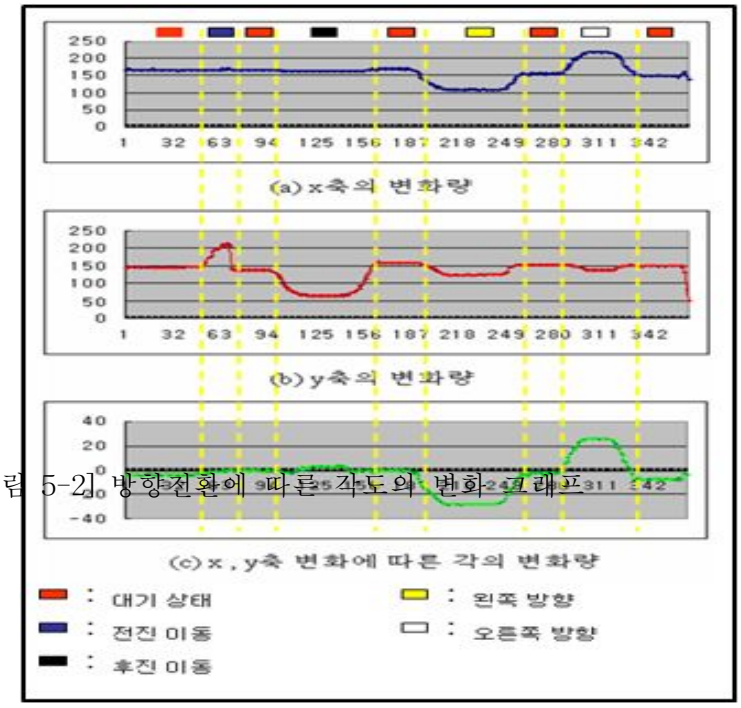
Collection @ chosun

1-1. 템플리트에 의한 인식률 및 실험 결과

실험을 통해서 인식률 및 방향전환을 확인하기 위하여 실험에서 필요로 하는 영상은 한 개의 손가락 끝 점을 가지는 손 형상이며, 전진, 후진, 좌측, 우측의 방향전환은 손 형상에서의 손가락 끝을 움직임으로써 전환을 한다. 전진, 후진은 손가락 끝 점의 위치 변화로 인식을 하고, 좌측, 우측은 손가락을 좌,우로 이동시킴으로써 방향을 인식하게 된다. [그림 5-2]는 [그림 5-1]에 보이는 5가지의 손 형상이 입력 정보로 들어 왔을 때 손가락 끝 점을 전진, 후진, 좌측, 우측으로 움직였을 때 변화되는 데이터들의 값을 그래프로 나타낸 것이다. (a)는 x 축 상에서 변화되는 값이고, (b)는 y 축 상에서 변화되는 값이며, (c)는 x, y 축의 변화에 따른 손가락 끝 점에 대한 각의 변화량이다.

[그림 5-1] 실험에 사용된 손 형상

(대기상태) (오른쪽) (왼쪽) (전진) (후진)



[그림 5-2] 방향전환에 따른 각도의 변화 그래프

Collection @ chosun

[표 5-1]에서는 [그림 5-3]에 보이는 것과 같이 10개의 손 형상을 사용하여 단순한 배경과 복잡한 배경으로 나눈 후, 손 형상에 대해 템플리트 매칭을 이용하여 손가락 끝 점의 인식에 따른 성공률과 실패율 결과를 수치로 나타낸 것이다. 한 개의 손가락 영상이 입력되었을 경우에는 높은 인식률을 보였지만, 다수의 손가락 영상이 입력되었을 경우에는 정확한 인식을 하기 어려웠다. 다수의 손가락 경우 손의 움직임이 수시로 변하기 때문에 템플리트 매칭의 결과 값도 수시로 변하게 됨으로 매칭 결과에 의한 손가락의 위치도 수시로 변하게 된다.

[표 5-1] 배경에 따른 손가락 끝 점 추출 인식률

배경 영상의 종류	영상의 개수	성공률(%)	실패율(%)
단순한 배경 (손가락 한 개)	5	98.2	1.8
복잡한 배경 (손가락 한 개)	5	96.5	3.5
단순한 배경 (다수의 손가락)	5	85.0	15.0
복잡한 배경 (다수의 손가락)	5	82.0	18.0

a. 한 개의 손가락 영상

[그림 5-3] 실험을 위한 10개의 손 형상

b. 다수의 손가락 영상

[표 5-2]는 [그림 5-3]의 10개의 손 형상을 단순한 배경과 복잡한 배경으로 나눈 후, 피부색 정보에 의한 손 영역 추출의 인식 결과를 수치로 나타낸 것이다.

Collection @ chosun

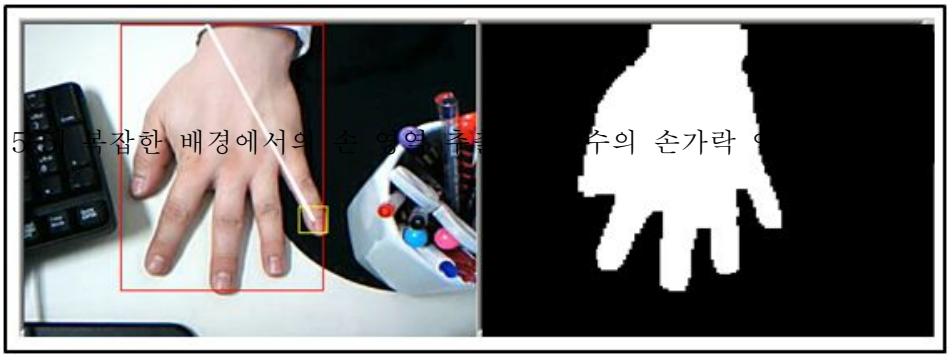
[표 5-2] 배경에 다른 손 영역 추출

배경 영상의 종류	영상의 개수	성공률(%)	실패율(%)
단순한 배경	10	98.2	1.8
복잡한 배경	10	96.5	3.5

[그림 5-4]는 복잡한 배경에서 한 손가락이 입력영상으로 입력되었을 경우 손 영역만을 추출한 후 추출된 손 영역에서 한 개의 손가락을 인식하는 것을 보여준다.

[그림 5-4] 복잡한 배경에서의 손 영역 추출 및 손가락 인식

[그림 5-5]는 복잡한 배경에서 다수의 손가락이 입력영상으로 입력되었을 경우 손 영역만을 추출한 후 추출된 손 영역에서 한 개의 손가락을 인식하는 것을 보여주고 있다. 하지만 한 개의 손가락을 찾기는 하지만 다수의 손가락이 있기 때문에 손이 움직일 때마다 매칭결과에 의한 손가락의 위치도 변하게 된다. 따라서 다수의 손가락이 입력되었을 경우에는 매칭 인식에 많은 어려움이 있다.

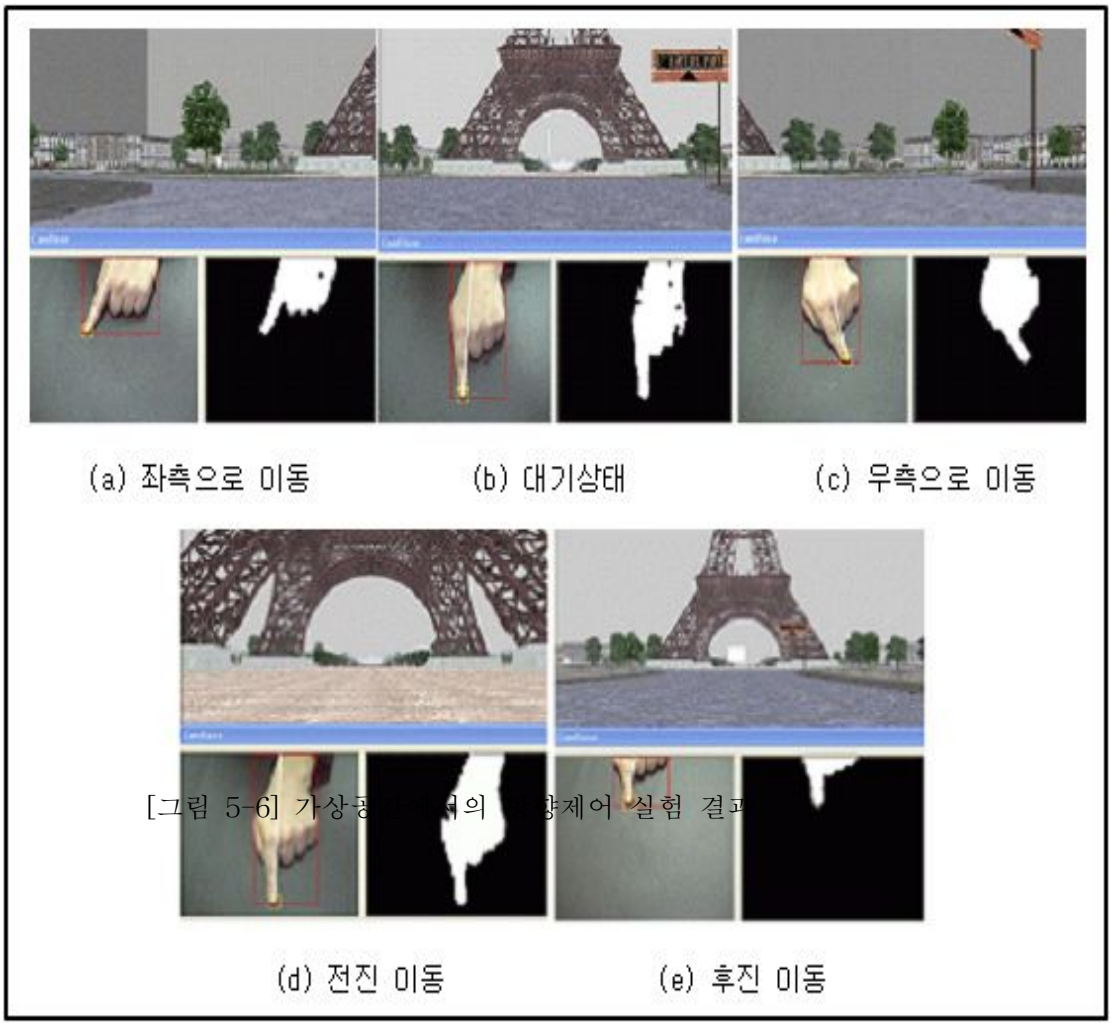


[그림 5-5] 복잡한 배경에서의 손 영역 추출 및 다수의 손가락 인식

Collection @ chosun

1-2. 가상환경에서의 응용 및 실험 결과

본 논문에서 제안하는 방법은 방향정보를 이용하는 Game 분야, 3D 가상환경에서의 Navigation 기능, Window System 제어 또는 하드웨어와 접목하여 가정용 가전제품을 제어하는 곳에 적용할 수 있으며 기타 많은 응용분야에도 적용할 수 있다. 본 논문에서는 손가락 끝 점을 이용하여 방향을 제어하기 위해 3D 가상환경 상에서의 Navigation에 적용하였으며, 전진, 후진, 좌측, 우측 등의 방향전환 명령이 효율적으로 사용될 수 있음을 확인하였다. [그림 5-6]은 가상공간에서의 Navigation이 작동되는 장면이다. (a)는 왼쪽으로의 이동, (b)는 초기의 대기화면(초기상태), (c)는 오른쪽으로의 이동, (d)는 대기상태에서의 앞으로의 이동, (e)는 대기상태에서의 뒤로 이동하는 장면을 나타내고 있다.



[그림 5-6] 가상공간에서의 방향제어 실험 결과

Collection @ chosun

[표 5-3]에서는 [그림 5-1]에 보이는 것과 같이 실제로 실험에 사용된 5개의 영상에 대해서 실험을 통해 확인된 방향전환 인식 결과를 나타낸 것이다.

[표 5-3] [그림 5-1]의 방향정보를 이용한 방향전환 인식 결과

영상의 종류	영상의 개수	성공률(%)	실패율(%)
단순한 영상	5	98.2	1.8
복잡한 영상	5	96.5	3.5

2. 신경망에 의한 손 형상 인식

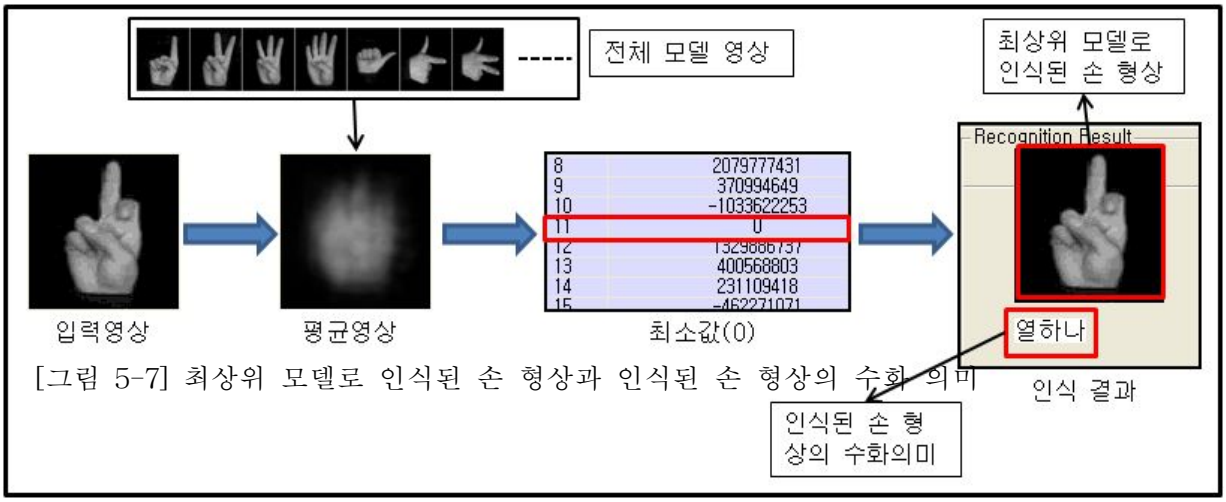
본 논문에서는 수화인식을 하기 위한 손 형상 인식 시스템을 구현하고자 한다. 수화를 인식하기 위해서는 손 형상을 인식해야 하는데 먼저, 손 형상을 정확하게 인식하기 위해 [그림 4-4]와 같이 모델 영상의 크기를 100*100으로 정규화했으며, 인식 속도를 높이기 위해 영상을 Gray Level로 변환하였다. 손 형상 인식에 사용되는 모델 영상은 총 100개의 영상으로 이루어져 있으나 본 논문에서는 일부의 모델 영상만을 담았다. 카메라로 들어오는 영상을 입력정보로 사용하였으며, 피부색 정보를 이용하여 영상의 전체 영역에서 손 영역만을 분리하였다. 분리된 손 영역으로부터 손 형상을 정확하고 안정적으로 인식하기 위해 SOFM(Self Organized Feature Map)이라는 신경망 알고리즘을 사용하였으며, SOFM 신경망 알고리즘을 사용함으로써 입력영상에 대한 인식률을 향상시켰다.

2-1. 신경망에 의한 손 형상 인식 결과

수화인식을 위한 손 형상 인식 시스템은 입력 영상을 전처리 과정에 의해 손 영역을 분할하고, 분할된 손 영역을 손 형상으로 인식한다. 입력 영상의 손 형상정보가 들어오면 100개의 전체 모델 영상에 대해 평균 영상을 구성한 DB(Database)[그림 4-21] 안에서 입력 영상의 특징 벡터와 모델 영상의 특징 벡터 사이의 거리를 SOFM 신경망 알고리즘의 경쟁과정[그림 4-22], 학습과정[그림 4-24], 적응과정[그

Collection @ chosun

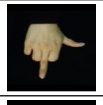
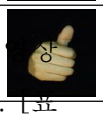
림 4-24]을 통해 비교하고, 거리가 최소값(0)이 되는 모델을 찾는다. 거리가 최소값이 되는 모델이 우리가 찾는 최상위 모델이 된다[그림 4-26]. 최상위 모델로 선택된 손 영상을 손 형상으로 인식한 후 인식된 손 형상을 시스템 상에서 다시 보여준다. 또한 100개의 전체 모델 영상에 각각 수화 의미를 부여하였으므로 인식된 손 형상을 보여줄 때 그 손 형상에 따른 수화의미도 같이 보여준다. [그림 5-7]는 입력 영상으로부터 모델 영상과의 거리를 비교하여 최소값(0)의 모델을 찾아 최상위 모델의 손 형상으로 인식하고, 인식된 손 형상에 따른 수화의미를 보여주고 있다.



[그림 5-7] 최상위 모델로 인식된 손 형상과 인식된 손 형상의 수화 의미

Collection @ chosun

[그림 5-8]은 손 형상을 인식하기 위한 100개의 전체 모델 영상에 따른 수화의미이다. 여기서는 영상의 일부만 보여준다.

	하나		열하나		제자리		사랑		미음
	둘		열둘		서기		관심		비읍
	셋		열셋		앞으로		행복		시옷
	넷		열넷		뛰기		미움		이응
	다섯		열다섯		위쪽		증오		지읒
	여섯		열여섯		점프		악마		치읓
	일곱		열일곱		아래쪽		천사		키읔
	여덟		열여덟		앉음		기역		티읕
	아홉		열아홉		오른쪽		디귿		피읖
	스물		열스물		왼쪽		비읍		히읗

[그림 5-8] 손 형상에 대한 수화의미

[표 5-4]는 손 형상에 대한 단순한 배경에서의 입력 영상과 복잡한 배경에서의 입력 영상에서 손 영역만을 추출하여 손 형상을 정확히 인식하는 비율을 나타낸 것이다. [표 5-4]에서의 인식비율을 보면 알 수 있듯이 단순한 배경에서의 손 영역을 추출하여 손 형상을 인식하는 것이 복잡한 배경에서의 손 형상을 인식하는 것보다 인식률이 좋은 것을 알 수 있다. 성공률은 손 형상을 정확히 인식한 결과이고, 실패율은 연결강도의 가중치가 최상위 모델에 가장 인접한 모델로 잘못 선택되면서 나타나는 현상이다. 또한 본 논문에서의 수화인식은 한 손으로만 수화를 표현한다는 가정으로 실험이 이루어졌으므로 두 손이 입력으로 들어왔을 경우는 손 형상으로 인식하지 못한다.

Collection @ chosun

[표 5-4] 단순한 배경과 복잡한 배경에서의 손 형상 인식률

배경 영상의 종류	영상의 개수	성공률(%)	실패율(%)
단순한 배경	100	98.2	1.8
복잡한 배경	100	95.5	4.5

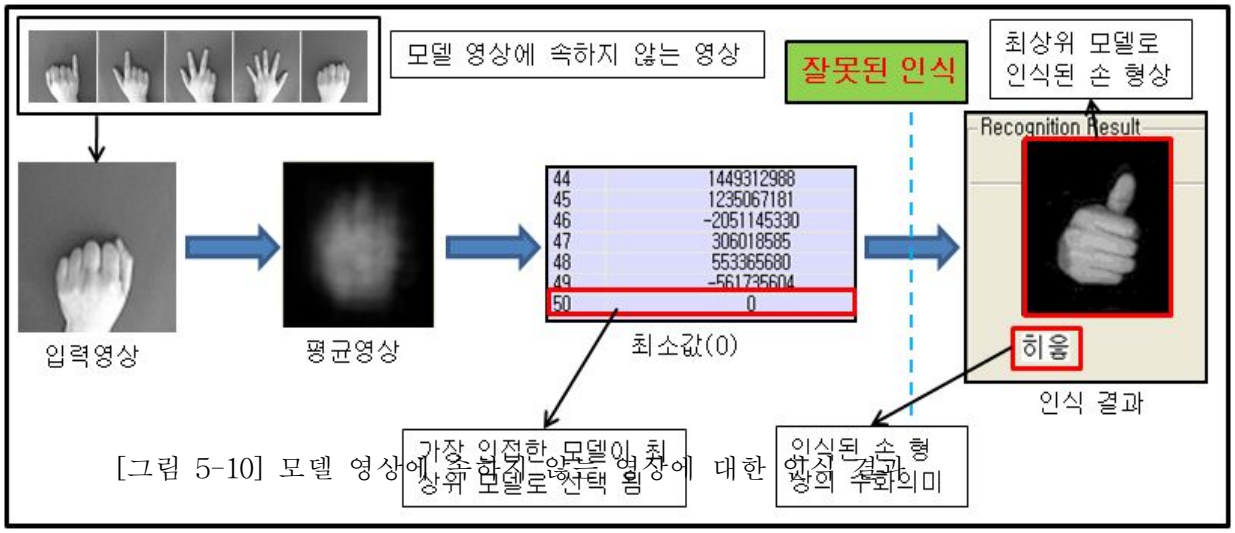
[표 5-5]는 손 영역에 대한 손 형상을 인식하는데 있어 인식하려는 입력 영상의 손 형상이 DB로 구성된 100개의 전체 모델 영상에 속할 때와 100개의 전체 모델 영상에 속하지 않을 때의 인식률을 나타낸 것이다. 모델 영상에 속할 때의 인식률은 98.8%로 매우 좋은 결과를 보이지만, 모델 영상에 속하지 않을 때의 인식률은 0%로 매우 나쁜 결과를 나타낸다. 이것은 연결강도의 가중치를 반복적으로 수행함으로써 최상위 모델을 찾아내고, 그 최상위 모델을 선택하여 손 형상으로 인식한다. 그러나 입력 영상으로부터 모델과의 거리가 최소값(0)이 되는 모델이 없으므로 입력 영상에 가장 인접한 모델이 최상위 모델로 선택이 된다. 따라서 선택된 최상위 모델과 입력 영상이 일치하지 않으므로 인식률이 0%가 된다. [그림 5-9]에서 보이는 5개의 손 형상은 100개의 전체 모델 영상에 속하지 않는 영상들이며, [그림 5-10]은 100개의 전체 모델 영상에 속하지 않는 영상이 입력 영상으로 입력되었을 때 손 형상에 대해 잘못 인식하는 결과를 나타낸다.

[표 5-5] 전체 모델 영상에 속하는 영상과 속하지 않는 영상의 인식률

모델 영상	영상의 개수	성공률(%)	실패율(%)
속하는 영상	100	98.2	1.8
속하지 않는 영상	20	0.0	100

[그림 5-9] 모델 영상에 속하지 않는 손 형상

Collection @ chosun



[그림 5-10] 모델 영상에 속하지 않는 영상에 대한 인식 결과

[표 5-6]은 본 논문에서 제안하는 SOFM신경망 알고리즘을 이용하여 손 형상을 인식했을 경우와 기존 다른 인식 알고리즘에 의해 인식된 경우와의 인식률 차이를 알아보기 위한 인식률 성능평가 분석표이다.

[표 5-6] 기존 알고리즘과 비교한 성능평가 분석표

알고리즘	모델 영상 분류	성공률(%)	실패율(%)
Pattern Matching 인식	속하는 영상	96.3	4.7
	속하지 않는 영상	0.0	100
PCA를 이용한 인식	속하는 영상	97.8	2.2
	속하지 않는 영상	0.0	100
SOFM 신경망을 이용한 인식	속하는 영상	98.2	1.8
	속하지 않는 영상	0.0	100

[표 5-6]의 각 알고리즘에 의한 인식률을 보면 알 수 있듯이 모델 영상에 속하는 영상의 경우 Pattern Matching을 이용한 인식률과 PCA를 이용한 인식률 그리고 본 논문에서 제안하는 SOFM 신경망을 이용한 인식률은 서로 근사한 인식률을 보이고 있지만 SOFM 신경망 알고리즘을 이용하여 손 형상을 인식한 경우가 PCA를 이용하여 손 형상을 인식한 경우보다 0.4% 더 좋은 인식 결과를 보였다. 하지만 모

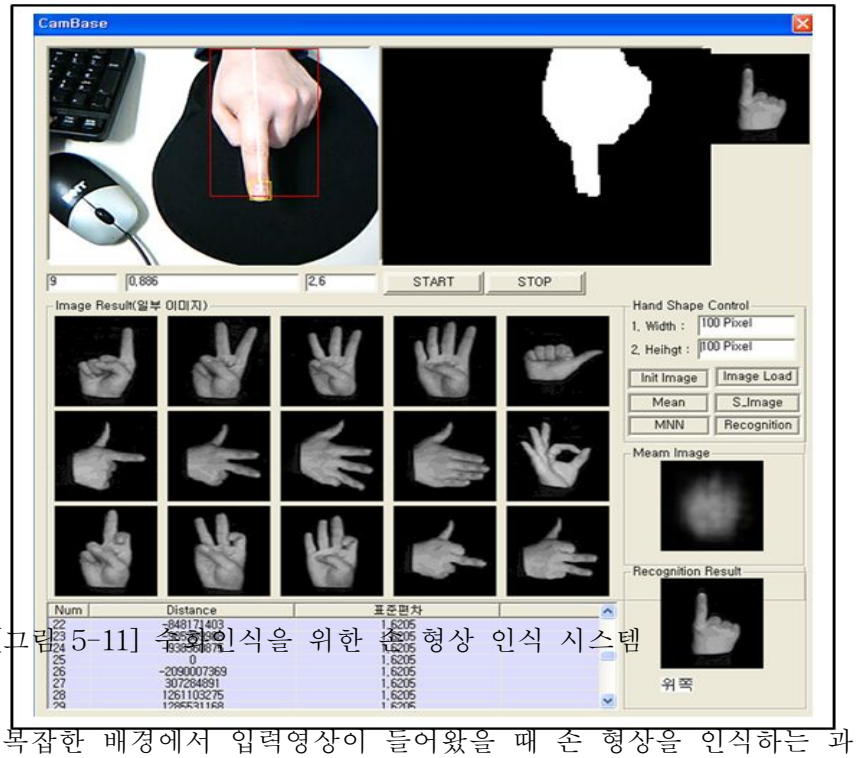
Collection @ chosun

델 영상에 속하지 않는 영상에 대해서는 적용된 알고리즘 모두의 인식률이 성공률(0)%, 실패율(100%)를 보였다. 이것은 모델영상 내에 입력영상이 없기 때문에 인식 과정에 있어 오-인식(Error Recognition)을 하게 되면서 나타나는 현상이다.

2-2. 손 형상 인식의 응용 및 실험 결과

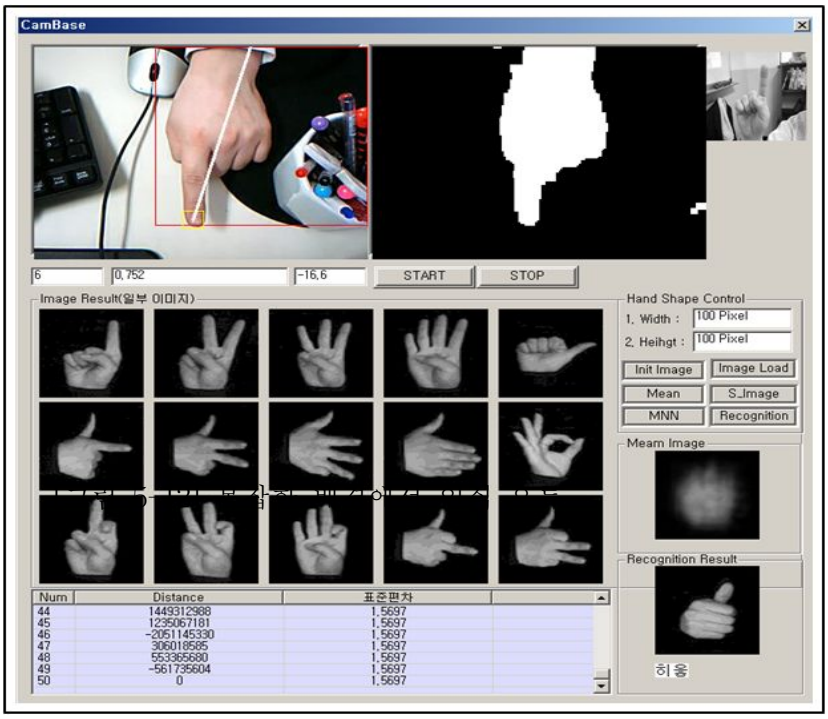
본 논문에서 제안된 시스템은 수화인식을 위한 손 형상 인식 시스템으로서 비장애인과 장애인 간에 원활한 의사 전달수단으로 유용하게 사용될 수 있다. [그림 5-11]부터 [그림 5-14]까지는 수화인식을 위한 손 형상 인식 시스템의 전 과정을 보여주고 있다. 손 형상 인식 시스템의 과정은 첫째, 카메라로부터 들어온 입력 영상의 전체 영역에서 피부색 정보를 이용하여 손 영역만을 추출한다(전처리 과정으로 100개의 전체 모델 영상에 대해 크기를 100*100으로 정규화하였으며, 연산속도를 줄이기 위해 모델 영상을 Gray Level(회색조) 정규화 시켰다). 둘째, 전체 영상으로부터 추출된 손 영역을 손 형상으로 정확하고 안정적으로 인식하기 위해 SOFM(Self Organized Feature Map) 신경망 알고리즘을 적용한다. 셋째, SOFM 알고리즘의 경쟁과정, 학습과정, 적응과정을 통해 입력 영상의 특징 벡터와 모델 영상의 특징 벡터의 거리를 계산한 후 초기 연결강도로부터 점차 연결강도의 가중치를 부여함으로써 입력 영상과 모델 영상간의 거리가 최소값(0)이 되는 모델을 찾고, 찾은 최소값의 모델을 최상위 모델로 선택하며, 최상위 모델로 선택된 손 형상을 인식한다. [그림 5-11]의 모델 영상은 시스템 내에 전체 모델 영상을 표시할 공간이 부족하여 일부 영상만 보여준 것이다.

Collection @ chosun



[그림 5-11] 손형 인식을 위한 손 형상 인식 시스템

[그림 5-12]는 복잡한 배경에서 입력영상이 들어왔을 때 손 형상을 인식하는 과정에서 오-인식(Error Recognition)을 했을 경우이다.



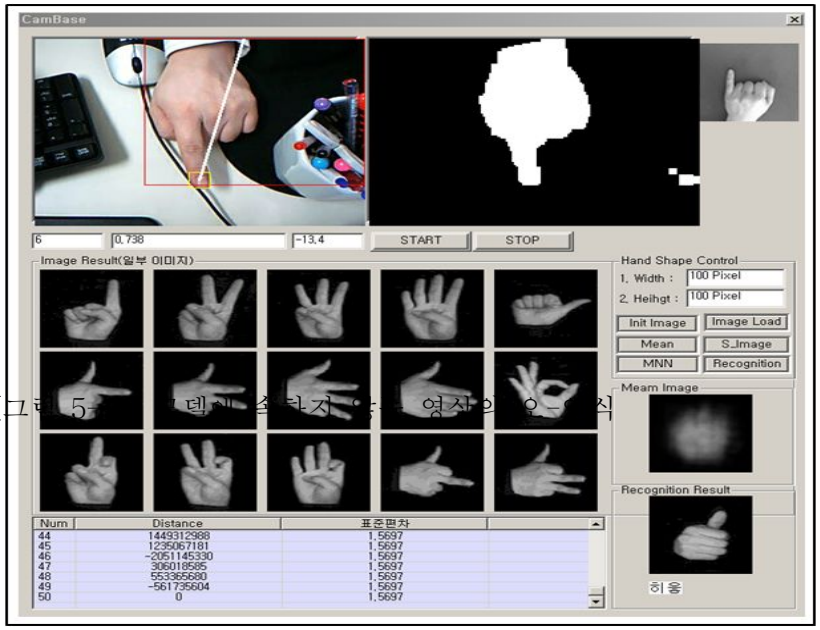
[그림 5-12] 복잡한 배경에서의 인식 오류

Collection @ chosun

[그림 5-13]은 복잡한 배경에서 입력영상이 들어왔을 때 손 형상을 정확히 인식하는 것을 보여준다.

[그림 5-13] 복잡한 배경에서의 인식

[그림 5-14]는 단순한 배경에서 모델영상에 속하지 않는 영상이 입력영상으로 들어왔을 때 손 형상 인식하는 과정에서 입력영상과 모델영상과의 거리가 가장 최소가 되는 모델을 선택하여 인식한 결과이다. 즉, 오-인식 결과이다.



[그림 5-14] 모델에 속하지 않는 영상의 오-인식

Collection @ chosun

VI. 결론 및 발전 계획

본 논문에서는 마우스, 키보드 등 기타 주변기기를 사용하지 않고 오직 사용자의 손가락을 이용하여 가상환경의 공간상에서 좌·우 방향전환 및 전·후 이동을 제어하는 시스템과 장애인과 비장애인 사이에 원활한 의사 전달을 위하여 수화를 인식하고, 인식되어진 수화의 의미를 전달하는 수화인식 시스템을 구현하였다. 본 실험에 사용된 하드웨어부분과 소프트웨어부분의 구성은 범용 USB 카메라(초당 30 프레임)를 장착한 Pentium IV 2.4GHz, Memory 1GB를 가진 PC환경에서 수행하였으며, 사용한 프로그램은 Visual C++를 사용하여 구현되었다.

전자의 경우, 사용자의 손가락을 이용하여 가상환경의 공간상에서 좌·우 방향전환 및 전·후 이동을 제어하기 위해서는 사용자의 손가락 정보를 알아야 한다. 손가락 정보를 알기 위해서는 카메라를 통해 들어오는 입력 영상의 전체 영역에서 피부색 정보를 이용하여 손 영역만을 추출하고, 추출된 손 영역에서 Morphology기법을 이용하여 잡음을 제거하였다. 추출된 손 영역에서 손가락을 정확히 인식하기 위해 손가락 끝 점을 템플리트 매칭으로 찾는다. 매칭 결과 매칭률이 가장 높은 모델을 찾아 손가락 끝 점으로 인식하고, 손가락 끝 점을 찾은 후 이 정보를 이용하여 전진, 후진, 좌측, 우측 등 사용자가 가리키는 방향을 인식하고, 인식된 방향 정보를 가상환경에서 인터페이스로 사용한다. 손가락 끝 점은 템플리트매칭을 이용하여 구하였으며, 손목의 중심과 손가락 끝 점까지를 기준 축으로 하여 손가락 끝 점을 전진, 후진, 좌측, 우측으로의 움직임 정보를 이용하여 방향정보를 추출한다. 또한 본 연구에서 제안하는 방법을 실험을 통해 확인하기 위해 3D 가상환경 공간에서의 방향전환 제어 실험을 한 결과 매우 좋은 인식률의 결과를 보였으며, 전진, 후진의 이동 및 좌측, 우측의 방향전환도 좋은 결과로 작동되는 것을 확인하였다.

후자의 경우, 장애인과 비장애인 사이에 원활하게 의사를 전달하기 위해 언어로 표현되는 수화를 인식해야 한다. 수화를 인식하기 위해서는 손 형상을 인식해야 하는데 전자의 경우와 같이 손 영역을 추출하여 잡음을 제거한 후 추출된 손 영역을 손 형상으로 인식한다(전처리 과정으로 100개의 전체 모델 영상에 대해 크기를 100*100으로 정규화하였으며, 연산속도를 줄이기 위해 모델 영상을 Gray Level 로 정규화 시켰다). 손 형상을 보다 정확하고 안정적으로 인식하기 위해서 SOFM 신

Collection @ chosun

경망 알고리즘을 사용하였다. SOFM 알고리즘의 경쟁과정, 학습과정, 적응과정을 통해 입력 영상의 특징 벡터와 모델 영상의 특징 벡터의 거리를 계산한 후 초기 연결강도로부터 점차 연결강도의 가중치를 부여함으로써 입력 영상과 모델 영상간의 거리가 최소값(0)이 되는 모델을 찾고, 찾은 최소값의 모델을 최상위 모델로 선택하며, 최상위 모델로 선택된 손 형상을 인식한다. 손 형상의 인식 정도를 알아보기 위해 모델DB에 속하는 영상이 입력으로 들어왔을 때와 모델DB에 속하지 않는 영상이 입력으로 들어왔을 때의 인식과정과 인식률을 실험을 통해 확인한 결과 모델DB에 속하는 영상일 경우는 매우 높은 인식률을 보였지만, 모델DB에 속하지 않는 영상일 경우에는 인식률이 0%로 매우 좋지 않은 결과를 보였다. 이것은 연결강도의 가중치를 반복적으로 수행함으로써 최상위 모델을 찾아내고, 그 최상위 모델을 선택하여 손 형상으로 인식하는데 있어 입력 영상으로부터 모델과의 거리가 최소값(0)이 되는 모델이 없으므로 입력 영상에 가장 인접한 모델이 최상위 모델로 선택되어졌다. 하지만 선택된 최상위 모델은 입력 영상과 일치하는 모델이 아니므로 인식률이 0%가 된다. 따라서 좀 더 많은 수화를 표현하기 위해서는 그 만큼 많은 모델 영상으로 모델DB를 구성하면 된다.

본 논문에서 제안하는 시스템은 범용 USB 카메라를 사용하기 때문에 저 비용 인식 시스템이라 할 수 있다. 또한 피부색 정보에 의한 손 형상 인식과 템플리트매칭을 이용한 손가락 끝 점 인식 방법은 수행속도가 매우 빠르며 안정적인 인식률을 보인다. 그러나 보다 효율적이고 안정적으로 수행하기 위해서는 현재 문제점으로 지적되는 조명변화와 복잡한 배경에서의 손 영역 추출 그리고 여러 개의 손가락이 입력되었을 때의 정확한 인식 등을 보완해야 한다. 현재 한 개의 손가락 끝 점을 갖는 손 형상만을 사용하였지만 향후에는 여러 개의 손가락 끝 점을 갖는 손 형상을 인식할 것이며, 조명변화에 강한 방법도 모색하여 인식률을 향상시킬 것이다. 또한 손 형상 인식에서도 신경망 알고리즘 외의 다른 알고리즘과 접목시켜 손 형상 인식에 더 좋은 결과를 낼 수 있도록 노력을 아끼지 않을 것이다. 이러한 시스템은 현재 3D 가상환경 상에서 방향전환 기능만을 실험을 통해 확인하였지만 향후에는 Game 분야, Window System 응용 및 임베디드(Embedded) 시스템을 기반으로 한 가정용 가전제품의 제어 등 여러 분야에도 응용될 수 있고, 수화인식 시스템에서는 수화를 인식하는 시스템이 활성화되면 기존의 비장애인이 수화를 배워야 한다는 경제적인 부담과 시간적인 부담감을 줄일 수 있으므로 보다 효율적이며 경

Collection @ chosun

제적인 시스템이라 할 수 있다. 또한 본 실험에서는 한 손에 국한해서 수화를 인식하고 그 의미를 표현했지만 향후에는 양 손 모두를 사용해서 시스템의 질적인 수준을 높이고 더 많은 양의 수화를 표현하는 시스템으로 발전해야 할 것이다.

Collection @ chosun

참 고 문 헌

[1] Ragael C. Gonzalez and Richard E. Woods "Digital Image Processing", Prentice Hall, 2002.

[2] J. Ohya and Y.. Kitamura, etc, "Real-Time Reproduction of 3D Human Images in Virtual Space Teleconferencing" in Proc. of '93 IEEE Virtual Reality Annual Int. Symp. pp.408-414, 1993.

[3] J. Segen and S. Kumar, "Shadow Gestures: 3D Hand Pose Estimation Using a Single Camera," CVPR99, vol. 1, pp. 479-485, Fort Collins, Colorado, June, 23-25, 1999.

[4] K. Imagawa, S. Lu and S. Igi, "Color-Based Hand Tracking System for Sign Language Recognition," Proceedings of International Conference on Automatic Face and Gesture Recognition, Killington, 1996.

[5] R. Cipolla, Y. Okamoto, and Y.Kuno,"Qualitative visual interpretation of 3D hand gestures using motion parallax," in Proc. of MVA'92, pp.477-482, 1992

[6] V. Pavlovic, R. Sharma, and T. Huang, "Visual Interpretation of Hand Gestures for Human Computer Interaction: A Review", IEEE Trans. on PAMI, Vol.19, No.7, pp.677-695, 1997.

[7] William T. Freeman and Michael Roth, "Orientation Histograms for Hand Gestures Recognition," in Course Note of SIGGRAPH'94, pp.M-1-M-23, 1994.

[8] C. Uras and A. Verri, "On the Recognition of the Alphabet of the Sign Language Through Size Function," in Proc. of 12th IAPR, Vol. 2, pp.334-338, 1994.

[9] J. Segen and S. Kumar, "Shadow Gesture: 3D hand Pose Estimation using a Single Camera", in Proc. of CVPR'99, pp.479-485, Colorado, 1999.

[10] A. Utsumi and J. Ohya, "Multiple Hand Gesture Tracking using Multiple Cameras", in Proc. of CVPR'99, pp.473-478, Colorado, 1999.

[11] Jilin Tu, Huang. T, Hai Tao "Face as mouse through visual face tracking", IEEE Computer and Robot Vision, 2005. Proceeding. The 2nd Canadian Conference on 9-11 May 2005.

Collection @ chosun

[12] Eun Yi Kim, Sin Kuk Kang, Keechul Jung, Hang Joon Kim "Eye mouse:mouse implementation using eye tracking", IEEE Consumer Electronics, 2005. ICCE. 2005 Digest of Technical Papers. International Conference on 8-12 Jan. 2005.

[13] Norris. G, Wilson. E "The Eye Mouse, an eye communication device", Bioengineering Conference, 1997, Proceedings of the IEEE 1997 23rd Northeast 21-22 May 1997.

[14] Ouchi. K, Esaka. N, Tamura. Y, Hirahara. M, Doi. M "Magic Wand: and intuitive gesture remote control for home applications", Active Media Technology, 2005.(AMT 2005). Proceedings of the 2005 International Conference on 19-21 May 2005.

[15] 최상배, 안성우, 한국수어의 이론, 서현사, 2003.

[16] 이찬수 외, "지문자를 포함한 연속된 한글 수화의 실시간 인식 시스템 구현", 대한전자공학회논문지, 98-35C-6-9, pp.76-87, 1998.

[17] 송효섭 외, "손의 형상과 움직임 방향 정보를 이용한 수화 인식", 정보과학회 논문지(B), 제 26권 6호, pp.804, 1999.

[18] 박광현, "모바일 수화 인식 시스템의 개선에 관한 연구", 전자공학회논문지, 2006-43SC-4-1, 2006.

[19] X. Zhu, J. Yang, and A. Waibel, "Segmenting Hands of Arbitrary Color," Proc Int'l Workshop on Automatic Face and Gesture Recognition, pp.446-453, 2000.

[20] 윤호섭 외, "손동작 제스처 인식을 이용한 윈도우 관리기 제어", 제9회 영상처리 및 이해에 관한 워크샵 발표논문집, pp.264-269, 1997.

[21] 송효섭 외, "Optical Flow를 이용한 손의 움직임 및 영역 검출", 한국정보과학회 '98 봄 학술 발표논문집(B), pp.642-644, 1998.

[22] 이형지 외, "수화용 손 제스처 인식을 위한 특징 벡터 추출", 제11회 신호처리 합동학술대회, pp.895-898. 1998.

[23] P. Hong, M. Turk and T. Huang, "Gesture Modeling and Recognition using Finite State Machines," Proc, Int'l Workshop on Automatic Face and Gesture Recognition, pp.544-549, 2000.

Collection @ chosun

[24] 조오용, 김형곤, 고성제, 안상철, “인터액티브 가상환경을 위한 손 제스처 인식 시스템”, 대한전자공학회 36-s(4), pp.70-82, 1999.

[25] K. Imagewa, S. Lu, and S. Igi. "Color-Based hands Tracking System for Sing Language Recognition," in Proc. of IEEE International Conference on Automatic Face and Gesture Recognition, Killington, pp.151-156, 1996.

[26] Richard J. and M. Ibrahim Sezan, "Video Background Replacement without A Blue Screen," ICIP99. 28A03.

[27] 한영환, “한글 수화의 실시간 인식 시스템의 구현,“ 2005.

[28] 조재현, 엄성용, “다중마커를 이용한 수화인식기법”

[29] 이칠우, “복수 모델영상의 상위도 패턴을 이용한 손 형상 인식,” 1995

[30] R. C. Gonalez, and R. E. Woods, Digital Image Processing, 2nd Edition, Prentice-Hall, pp.265-302, 2002.

[31] Patrick K. Simpsion, "Fuzzy Min-Max Neural Networks - Part 1: Classification," IEEE Trans. on Neural Network, Vol. 3, No. 5, 1992.

[32] 이태엽, 이상완, 변증남, “ 퍼지 최대-최소 신경망을 이용한 특징 집합 선택에 관한 연구 및 보행 단계인식에의 응용,” Proceedings of KFIS Autumn Conference 2007 Vol. 17, No. 2.

[33] Rowley, Baluja, and Kanade: Neural Network-Based Face Detection (PAMI,January 1998).

[34] Robert Brunelli and Tomaso Poggio, "Face Recognition : Feature versus Templates," PAMI, Vol, 15, no. 10, pp.1042-1052, Oct, 1993.

[35] 송홍복, 설지환, “다층신경망을 이용한 임의의 크기를 가진 얼굴인식에 관한 연구,” 퍼지 및 지능시스템학회 shsasn지 2005. Vol. 15, No. 2, pp.149-162.

[36] T. Kohonen, "Self-Organization and Associative Memory," Springer Verlag, 1984.

[37] 진성일, 김인철, 배건성, “Backpropagation Network의 Auotassociation Memory 기법을 통한 패턴 인식에 응용,” 전자공학회 Vol. 11. 1990.

[38] M. Watanabe, K, Kuwata and R. Katayma, "Adaptibe Tree-Structured Self Generting Radial Basis Function and its Application to Nonlinear Identification Problem," in Proc. IIZUKA, pp.167-170. 1994.

[39] 이동규, 이기정, 황보택근, 임혁규, “신경망을 이용한 동작분석과 원격

Collection @ chosun

응급상황 검출 시스템", 한국콘텐츠학회논문지, 제6권 제9호, pp.50-59, 2006.
[40] 김혜정, 이경미, "RBF 신경망을 이용한 실루엣 기반 유아 동작 인식", 한국인터넷정보학회, 8권 3호, 2007.
[41] T. Kohonen, J. Hunninen, J, Kangas, J. Kaaksonen, K. Torkkola, "LVQ_PAK: The Learning Vector Quantization Program Package.", Technical Report A30, Helsinki Univ.,1996.
[42] 권오륜 외, "부분공간과 LVQ 분류기에 기반한 실시간 얼굴 인식", 한국인터넷정보학회, 제8권 제3호, 2007.
[43] T. Kohonen, "Self-Organizing Maps," Springer Series in Information Science. Vol. 30, Springer, Berlin, Heidelberg, New York, 1995, 1997, 2001, Third Extended Edition
[44] D Chi, King N. Ngan "Face Segmentation Using Skin-Color Reference map in Videophone Applications", IEEE Transactions on Circuits and Systems for Vidoe Technology, vol.9, June 1999
[45] Ajmend Elgammal, David Harwood and Larry Davis, " Non-parametric Model for Background Subtraction", ECCV 2000, pp. 751-767
[46] Christopher Eveland, Kurt Konolige and Robert C. Bolles, "Background Modeling for Segmentation of Video-Rate Stereo Sequences", CVPR 1998, pp.266-271
[47] William T. Freeman and Michael Roth, "Orientation histograms for hand gesture recognition,"Intl. Workshop on Automatic Face and Gesture Recognition, IEEE Computer Society, pp. 296-301, June 1995, Zurich, Switzerland
[48] S. Ahmad, "A Usable Real-Time 3D Hand Tracker," Conference Record of the Asilomar Conference on Signals, System and Computers, pp. 1257-1261, 1994.
[49] Tony Heap and Ferdinando Samaria." Real-time hand tracking and gesture recognition using Smart Snakes," Proc. Interface to Human and Virtual Worlds, June 1995, Montpellier, France.