

(論 文)

# 초상화로봇을 위한 표정변환 및 드로잉규칙 생성

김문상\* · 민선규\* · 최창석\*\*

(1994년 2월 24일 접수)

## Facial Expression Transformation and Drawing Rule Generation for the Drawing Robot

Mun-Sang Kim, Sun-Kyoo Min and Chang-Seok Choi

**Key Words :** Facial Expression Transformation(표정변환), Deformation Rule(변형규칙), Standard Wire Frame Model(표준 와이어프레임모델), Texture Mapping(영상사상), Mask(마스크), Contour(윤곽), Drawing Path(그림경로)

### Abstract

This paper presents a facial expression transformation algorithm and drawing rule generation algorithm for a portrait drawing robot which was developed for the '93 Taejeon EXPO. The developed algorithm was mainly focused on the robust automatic generation of robot programs with the consideration that the drawing robot should work without any limitation of the age, sex or race for the persons. In order to give more demonstratin effects, the facial expression change of the pictured person was performed

### 1. 서 론

컴퓨터와 광전자학의 발달로 말미암아 근래 영상 처리 분야가 많이 발달되었으며, 이에 따라 영상처리에 필요한 많은 이론들이 쏟아져 나오고 있으며 그 적용분야도 점점 넓어가고 있다. 그 대표적인 적용분야로서는 측정과 인식이라고 할 수 있으며 많은 시스템들이 만들어져 사용되고 있다.

그러나 이러한 시스템들은 인공적인 대상물의 측정과 인식에 사용되며 많은 제한조건들 안에서 만족이 된다. 그리고 이러한 대상의 폭을 인간에게까지 확대하기 위한 많은 연구가 진행되고 있다. 그러한 연구를 크게 나누어 보면 인간얼굴의 인식, 표정변환, 모타주작성 등이 있으며 이때 가장 문제가 되는 것은 인간의 얼굴은 영상처리에 필요한 각종

변수가 명확하지 않다는 것과 얼굴의 요소에 대한 인식이 매우 어렵다는 것이며 이의 해결을 위하여 많은 연구가 현재 진행중이나 현재와 같은 알고리즘만으로는 해결하기 어려운 부분이 매우 많다.

본 연구에서는 '93 대전 엑스포에 출품된 초상화로봇 제작을 위하여 사람의 얼굴에 대한 표정변환과 윤곽선 추출에 대한 연구를 수행하였으며 그 결과 다수의 불특정한 인물에 대한 표정변환과 윤곽선 추출을 할 수 있게 되었다. 본 연구에서는 안정성에 가장 큰 주안점을 두었으며 이에 따라 요소인식과 같은 불안정한 결과를 낳는 처리는 되도록 채용하지 않았으며 각종 처리의 평가에 있어서 수학적 명확한 처리보다는 처리결과와 납득성에 큰 비중을 두었다. 즉, 인간적인 기준에 맞추어 많은 결과를 평가하였으며 이는 몇개의 파라미터만을 만족시킴으로써 얻을 수 있는 것은 아니었다. 그리고 처리의 신속성을 위하여 필요한 정보를 상당수 사전에 생성시켜 두어 그때그때 약간의 변형만은 주

\*정회원, 한국과학기술연구원 기전연구부

\*\*정회원, 명지대학교 정보통신공학과

는 방법을 사용하였으며 이는 안정된 결과의 생성에 큰 작용을 하였다.

## 2. 본 론

### 2.1 표정변환

표정변환은 얼굴의 정면영상을 획득한 후 이 영상을 몇가지 처리를 거쳐 각종의 표정 예를 들면 웃거나 찡그리는 표정을 만들어 내는 것으로서 만들어낼 수 있는 표정은 무한에 가깝다. 표정변환은 표준와이어프레임의 생성, 특징점의 추출, 표준와이어프레임의 핏팅, 변형규칙의 설정, 영상사상의 과정을 거쳐 수행이되며 본래 얼굴은 3차원이나 사진을 이용하여 처리하기 때문에 2차원적인 처리가 수행되었다.

#### 2.1.1 표준 와이어프레임의 생성

표정변환을 하기 위해서는 우선 변형규칙을 적용하기 위하여 표준 와이어프레임모델을 제작하는 것이 필요하다. 표준 와이어프레임의 형태는 삼각형의 패치로 제작이 되며 삼각형의 개수는 많을수록 좋으나 일반적으로 469개의 데이터 포인트와 이것들을 연결하는 496개의 삼각형으로 구성이 되며 그 모양은 평균적인 얼굴의 모양과 유사하도록 만든다. 본 시스템에서 사용한 표준 와이어프레임의 모양은 Fig. 1과 같다.

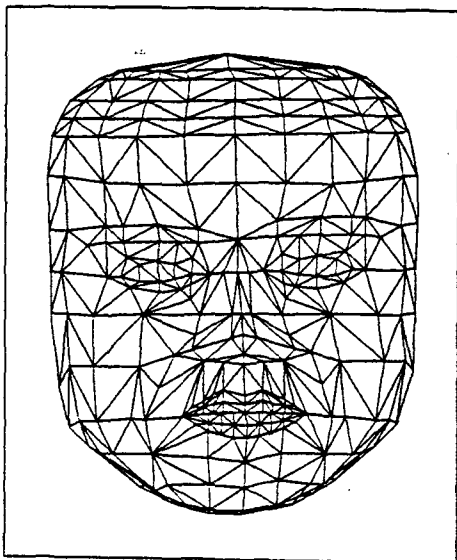


Fig. 1 Standard wire frame model

### 2.1.2 특징점의 추출

제작된 표준 와이어프레임을 그대로 사용할 수는 없으므로 이 표준 와이어프레임을 표정변화시키고자 하는 사람의 얼굴모양에 맞추어야 한다. 이 과정에서 필요한 것은 얼굴의 위치와 각 얼굴요소의 위치가 된다. 즉 눈, 코, 눈썹, 입의 위치와 모양에 대한 정보가 필요하다.

이 특징점을 추출하는 방법으로는 여러가지 방법이 존재한다. 그러나 대부분의 방법들이 전제조건을 요구하고 있으며 본 시스템과 같이 다양한 인종을 대상으로 하여 강인한 방법이 존재하지 않으므로 이 부분은 대화식으로 입력하도록 하였다.

그리고 여러 실험을 거친 결과 표정의 변화에 주로 작용하는 얼굴의 요소는 눈과 입이고 코나 눈썹은 그리 큰 영향이 없는 것으로 판명이 되어 코와 눈썹은 별도의 핏팅이 필요없이 전체적인 스케일링을 맞추는 것으로도 충분하였다. 그리하여 입력시 코와 눈썹에 대한 정보는 입력하지 않아도 무방하였으므로 입력되는 포인트의 수를 최소화할 수 있었다. Fig. 2에 필요한 특징점의 위치가 표시되어 있다.

### 2.1.3 표준 와이어프레임의 핏팅

표준 와이어프레임을 추출된 특징점을 이용하여 획득된 원영상에 맞추는 작업을 수행한다. 맞추는 작업은 크게 스케일링을 포함한 전체핏팅과 얼굴요

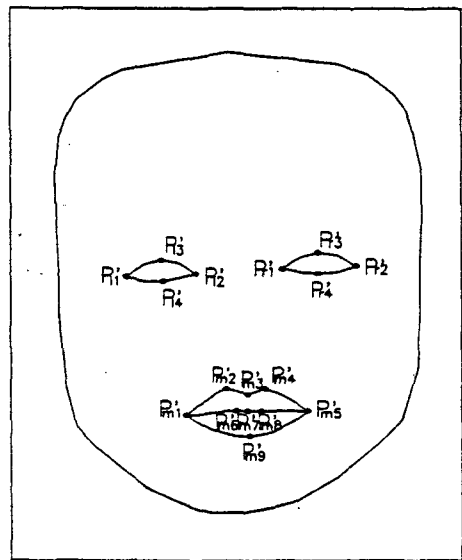


Fig. 2 Position of the characteristic point

소의 위치를 세밀하게 맞추는 부분핏팅만으로 나눌 수 있다. 전체핏팅을 할 때 주의해야 할 사항은 얼굴영상회특시 정면모습을 얻는 것이 매우 어렵고 항상 얼굴이 기울어져 있다는 사실이다. 그러므로 얼굴의 회전을 약간 고려해 주어야 한다.

Fig. 3에서와 같이  $\omega_x$ ,  $\omega_y$ ,  $\omega_z$ 의 회전이 있다. 그 중에서  $\omega_x$  방향의 회전은 일반적으로 매우 작아서 보정이 필요없으나  $\omega_y$ ,  $\omega_z$ 의 두 방향은 보정을 해 주어야 한다.

우선 전체핏팅을 하는 과정은 눈과 눈 사이의 점과 입술의 아래부분의 점 사이의 거리(OC)를 이용하여 수직방향의 스케일을 맞추고 다시 거리 OA와 OB를 이용하여 왼쪽면과 오른쪽면을 따로따로 핏팅하여 준다. 그 이유는 얼굴정면의 영상을 정확하게 획득하였다고 보기 어렵기 때문에 어느 정도 회전한 것( $\omega_y$ )을 보상하여 주기 위함이다. 그리고  $\omega_z$  방향의 회전도 또한 보상을 해 주어야 한다.

전체핏팅이 끝난 후 눈과 입에 대한 부분핏팅을 하게 된다. 그런데 얼굴요소의 점 수에 비해 핏팅해야 할 점이 많으므로 나머지 점들은 어핀 변환 (affine transform)을 하게 된다.

$$P_{3xn} = \begin{bmatrix} x'_1 & x'_2 & \dots & x'_n \\ y'_1 & y'_2 & \dots & y'_n \\ 1 & 1 & \dots & 1 \end{bmatrix} \quad (1)$$

$$P_{3xn} = \begin{bmatrix} x_1 & x_2 & \dots & x_n \\ y_1 & y_2 & \dots & y_n \\ 1 & 1 & \dots & 1 \end{bmatrix} \quad (2)$$

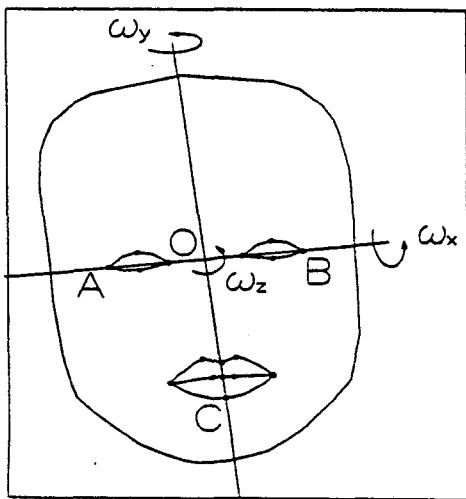


Fig. 3 Global fitting

여기서,  $P'$ 는 특징점의 위치이고  $P$ 는 전체핏팅을 거친 와이어프레임의 해당점의 위치,  $n$ 은 변환을 구하는데 참여하는 특징점의 수이다.

$$\begin{aligned} P' &= TP \\ P'P^T &= TPP^T \\ T &= (P'P^T)(PP^T)^{-1} \end{aligned} \quad (3)$$

이렇게 해서 구해진 변환행렬  $T$ 를 이용하여 부분변환을 하는데 이때 변환행렬  $T$ 는 왼쪽 눈, 오른쪽 눈, 입에 대해서 각각 구하여 진다. 그리고 눈의 경우 표준 와이어프레임을 보면 8개의 점으로 구성이 되나 변환시는 주위의 점 6개를 더 변환시켜주는데 이것은 과도한 변환이 발생시 와이어프레임의 겹침을 방지하기 위한 것이다. Fig. 4에 부분핏팅을 하는 점들이 도시되어 있다.

2.1.4 변형규칙의 성질과 변형량 결정

얼굴의 변형종류는 44가지로 나누어 볼 수가 있다. 이 변형은 Table 1과 같이 될 수 있다. 그리고 이 변형들은 해당 와이어프레임을 어느 정도 변형시키는데 대한 정보를 가지고 있게 된다. 즉 변형규칙을 44세트룰 가지고 있는 셈이 되며 이것들에 가중치를 두어 중첩시키게 되면 주어진 변형을 얻을 수 있게 된다. Fig. 5에 눈썹과 입에 대한 변형의 예가 있다.

Table 1의 변형규칙은 일반적으로 여러가지를

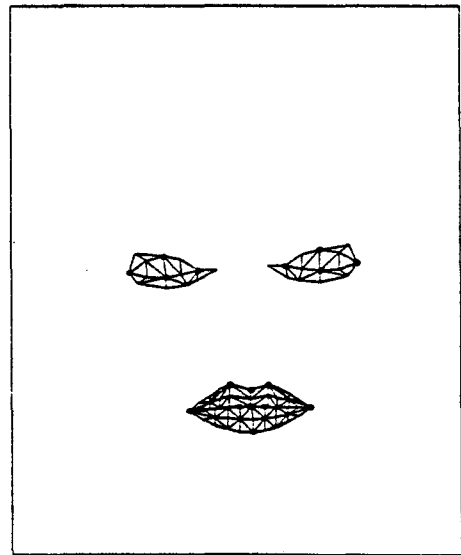


Fig. 4 Local fitting

중첩하여서 사용하여 여러가지 표정을 만들어 내게 되며 Fig. 6에 변형규칙을 어떻게 적용하는가가 나타나 있다.

이를 수식적으로 표현하여 보면 다음과 같다.

$$P_{(after\ deformation)} = \sum_{i=1}^N w_{(deformation\ rule)i} P_{(before\ deformation)i} \quad (4)$$

여기서,  $P_i$ 는 와이어프레임의 데이터 포인트이고  $w_i$ 는 변형규칙  $N$ 은 변형규칙의 수이다.

변형규칙은 일반적으로 고정시켜 놓으나 가중치는 어떤 표정을 만드느냐에 따라 다르게 설정될 수 있다. 그리고 같은 가중치를 적용한다 하더라도 사람에 따라서 결과적으로 변환된 얼굴이 주는 감정은 상당히 다를 수 있다. 그러므로 가중치를 정하는 작업은 자기가 만들고 싶은 얼굴을 설정하여 놓고 정하는 것이 보통의 수순이나 본 시스템에서는 그런 종류의 입력을 없애기 위하여 미리 약 30종류의 가중치 집합을 선정하여 놓고 무작위로 그 중의 한 변환이 일어나도록 하였고 때문에 결과에 대한 대략적인 추측만이 가능하다. 이렇게 한 이유는 시스템이 빠른 처리와 가능한 한 높은 정도의 자동화, 용이한 조작성을 원하였기 때문이다.

2.1.5 변환영상의 획득

위의 절차를 거쳐 표정변환된 와이어프레임을 얻을 수 있다. 이로부터 표정변환된 영상을 얻어야 하는데 이를 위해서는 Fig. 7과 같은 방법을 사용

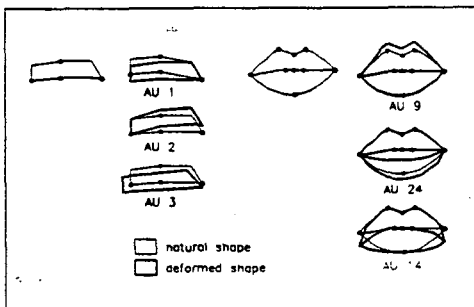


Fig. 5 Deformation example of the brow and mouth

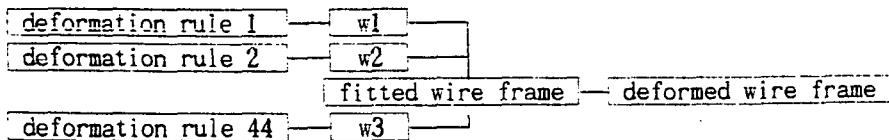


Fig. 6 Application of the deformation rule

하게 된다.

$$P'_i = s_i(P_b - P_a) + t_i(P_c - P_a) + P'_a$$

$$P_i = s_i(P_b - P_a) + t_i(P_c - P_a) + P_a \quad (5)$$

$P_i$ 의 색깔 =  $P'_i$ 의 색깔

위의 식을 이용하여 변형된 와이어프레임 상의

Table 1 Deformation rules

#	Rule	#	Rule
1	Inner brow raiser	23	Lip pressor
2	Outer brow raiser	24	Lips parts
3	Brow lower	25	Jaw drop
4	Upper lid raiser	26	Mouth stretcher
5	Cheek raiser	27	Lips suck
6	Lid tightener	28	Jaw thrust
7	Lips toward	29	Jaw sideways
8	Nose wrinkler	30	Jaw clencher
9	Upper lip raiser	31	Bite
10	Nasolabial furrow deepener	32	Puff
11	Lip, Corner puller	33	Cheek suck
12	Sharp lip puller	34	Tongue bulge
13	Dimpler	35	Lip wipe
14	Lip corner depressor	36	Nostril dilater
15	Lower lip depressor	37	Nostril compressor
16	Chin raiser	38	Nostril compressor
17	Lip pucker	39	Slit
18	Tongue show	40	Eyes closed
19	Lip stretcher	41	Squint
20	Neck tightener	42	Blink
21	Lip funneler	43	Blink
22	Lip tightener	44	Wink

한 점이 변형전의 와이어프레임상의 어느 점인가를 알 수 있게 되므로 이것을 전 포인트상에 대하여 수행하게 되면 색정보를 모두 알 수 있게 되므로

새로운 사진을 구성할 수 있게 된다.

이와 같이 함으로써 주어진 영상을 표정변환시킬 수 있게 되며 표정변환의 예가 Fig. 8에 나타나 있으며 Fig. 8에서 Au는 변형규칙을 지칭하는 말로

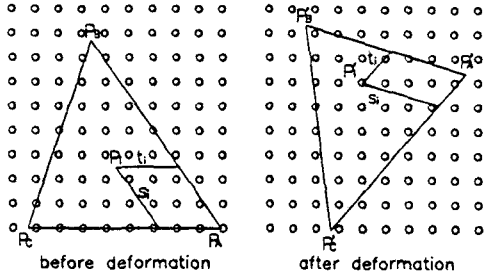
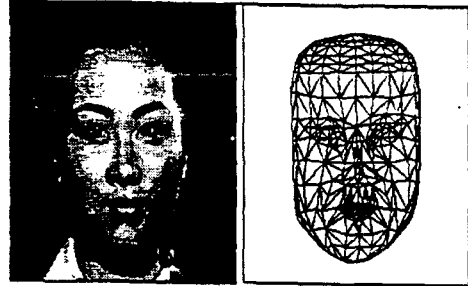
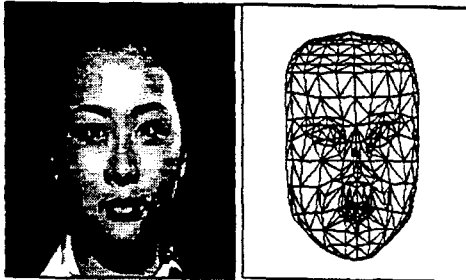


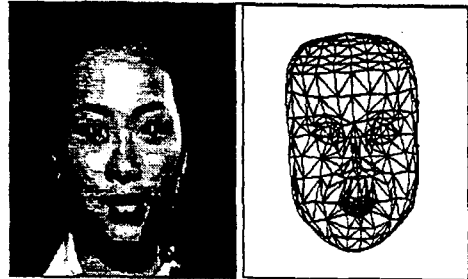
Fig. 7 Texture mapping



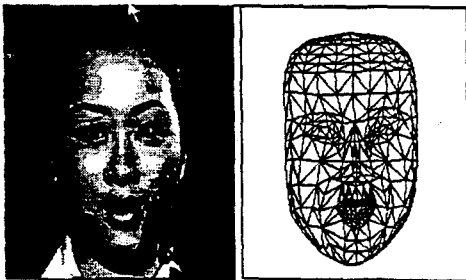
(d)  $0.2(Au5)+0.2(Au11)$



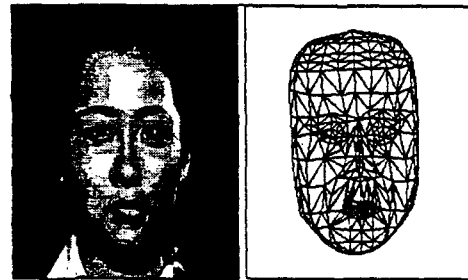
(a) Original image



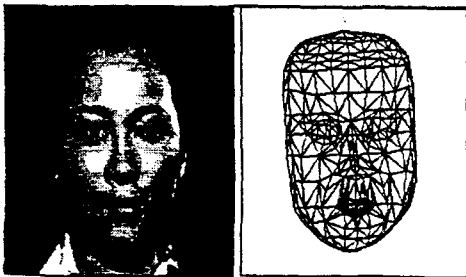
(e)  $0.2(Au5)+0.2(Au11)+0.2(Au24)$



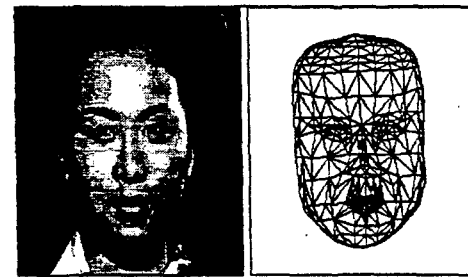
(b)  $0.59(Au1)+0.5(Au2)+0.2(Au5)+0.2(Au11)+0.2Au25$



(f)  $0.5(Au3)+0.2(Au8)+0.2(Au16)$

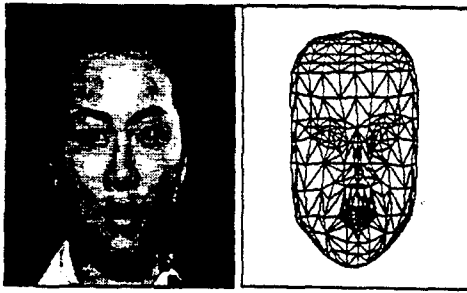


(c) Au44

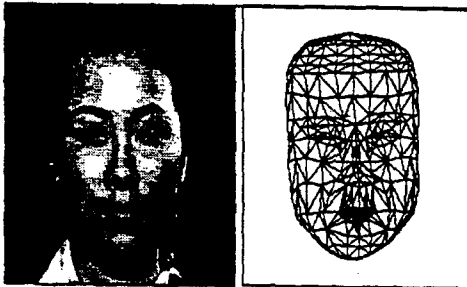


(g)  $0.5(Au3)+0.2(Au8)+0.2(Au8)$

Fig. 8 Example of the facial expression transformation



(h)  $0.1(Au3)+0.1(Au4)+0.1(Au6)+0.1(Au9)+0.1(Au25)$



(i) Au42

Fig. 8 Example of the facial expression transformation

서  $Au_i$ 은 병형규칙  $l$ 을 말하는 것이다.

2.2 인물사진으로부터의 윤곽선의 추출

표정변환이 되거나 또는 되지 않은 원 영상으로부터 실제로 로봇으로 하여금 그림을 그릴 수 있도록 윤곽정보를 추출하는 과정으로서 세 종류의 필터링결과를 영역별로 합성하는 과정으로써 대략적인 순서는 Fig. 9와 같다.

2.2.1 윤곽의 검출(결과 1, 2의 검출)

얼굴영상의 윤곽선을 찾아내기 위해 라플라시안 필터 중에서 엘리먼트의 합이 0인 ( $\sum a_{ij}=0$ ) 필터를 사용하였으며 필터의 크기는  $3 \times 3$ ,  $5 \times 5$ ,  $7 \times 7$ ,  $9 \times 9$ ,  $11 \times 11$ 의 다섯종류가 있으나 주로 사용한 것은  $3 \times 3$ 과  $7 \times 7$ 을 사용하였으며 Fig. 10에  $3 \times 3$ 과  $7 \times 7$  필터의 모양이 나타나 있다.  $3 \times 3$ 은 적용영역이 좁으므로 세밀하게 변화하는 윤곽을 찾는데 용이하므로 주로 얼굴내면의 묘사에 사용되었고  $7 \times 7$ 은 얼굴외면의 머리카락 등의 묘사에 주로 사용되

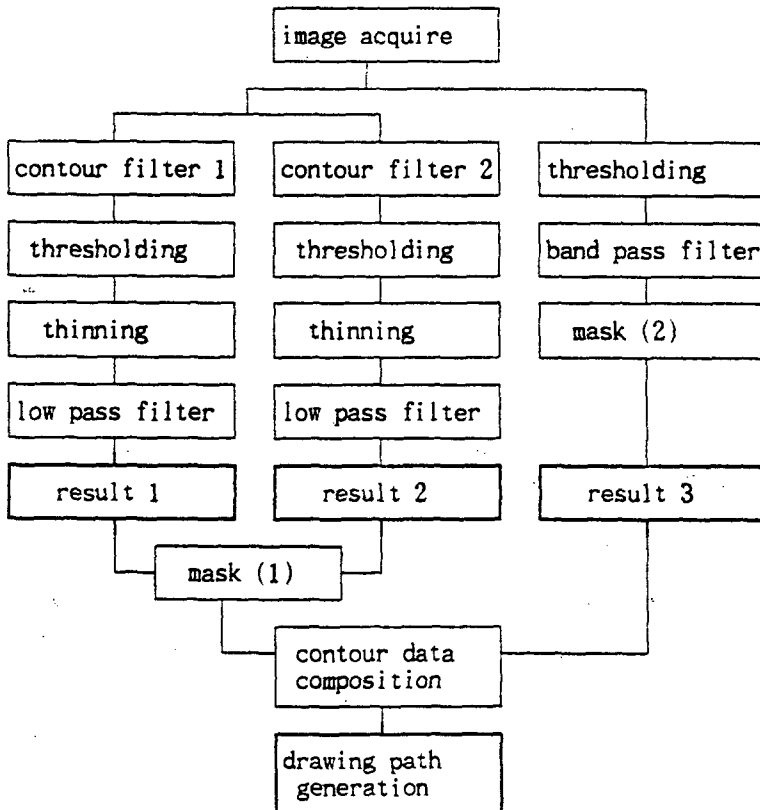


Fig. 9 Procedure for the extracting the contour from the original image

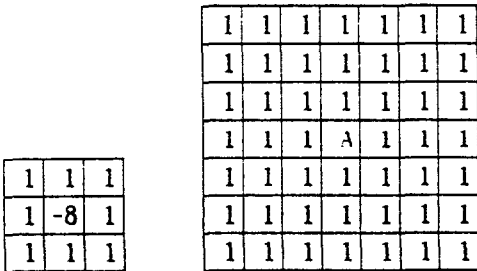
었다.

수식적으로 표현하면 다음과 같이 된다.

$$I_{ij} = \sum_{k=1}^{h=size-1} \sum_{l=0}^{l=size-1} a_k I_{i-k-size/2, j-l-size/2} \quad (6)$$

여기서,  $I_{ij}$ 는 영상처리결과이고  $I_{ij}$ 는 원영상의 밝기,  $size$ 는 필터의 크기이다.

이 필터를 사용한 결과를 이치화 및 세선화를 거쳐 결과 1과 2를 만든다. 그리고 원영상을 그대로 이치화시켜서 만든 결과 3을 얻는다. 그리고 이 결과 1과 2를 합성하는데에는 마스크를 이용하는데 이 마스크는 평균적인 사람의 얼굴내면을 정의하고 있으며 얼굴의 종류에 따라 Fig. 11와 같은 4종류의 마스크를 이용하고 있다. 4종류의 마스크를 이



3x3 filter

7x7 filter (A=-48)

Fig. 10 Used filter

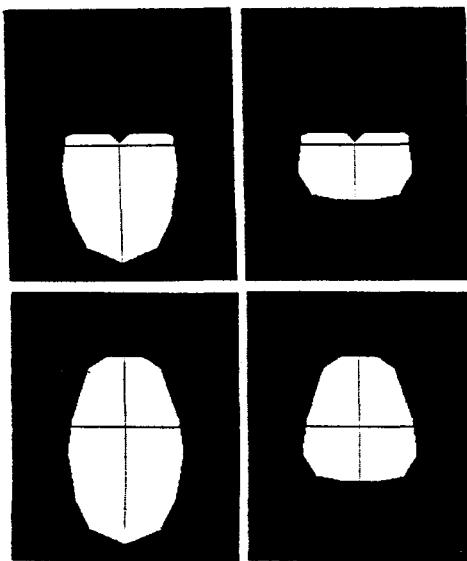


Fig. 11 Contour masks

용하는 것은 헤어스타일이나 턱수염의 존재유무때문이다.

2.2.2 얼굴요소의 검출(결과 3의 검출)

원영상을 이치화시킨 후 다시 대역필터(밴드패스 필터)를 거쳐서 얼굴요소(눈, 코, 입, 눈썹)의 넓이와 비슷한 부분만을 남겨 놓은 후 얼굴내면을 정의하는 마스크를 이용하여 얼굴내부의 요소를 얻는다. 이 마스크에는 Fig. 12와 같이 네 종류가 있는데 윤곽검출의 경우와 마찬가지로 헤어스타일이나 수염의 유무에 따라 마스크의 종류가 결정된다.

2.2.3 윤곽정보의 합성 및 드로잉경로 생성

결과 1, 2, 3을 모두 합성하여 하나의 데이터를 만든다. 그러므로 이 데이터에는, 얼굴의 윤곽이나 얼굴요소의 정보도 같이 들어가 있다. 이 정보로부터 드로잉경로를 생성하는데는 8방향의 벡터를 사용한다. 윤곽선의 끝을 시작으로 시계방향으로 벡터를 이용하여 다음 진행방향을 찾는다. 그리고 같은 진행방향이 계속될 경우에는 진행방향의 처음점과 끝점만을 연결하여 로봇의 이동경로를 간단하게 해 주었다. 이 알고리즘은 체인코드를 이용한 에지 디텍션(edge detection)과 유사한 면이 있다. 이치화를 통하여 얻어지는 결과 3에 대한 드로잉경로의 생성도 마찬가지로 적용이 된다. 윤곽으로부터 드로잉경로를 생성하는 예는 Fig. 13과 같다.

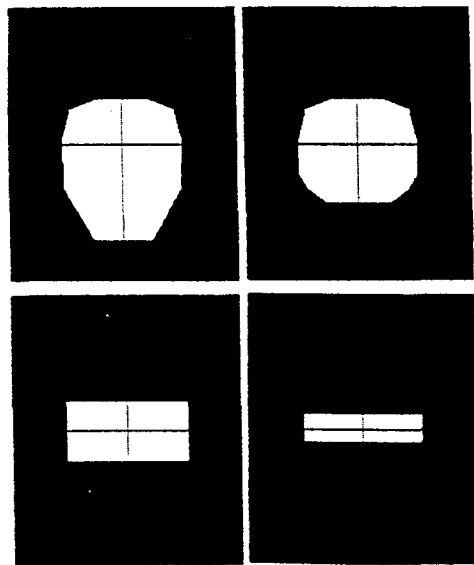


Fig. 12 masks for the finding the feature element

드로잉패스를 찾는 과정

스텝 1 : 스캐닝을 통하여 점 4를 찾는다

스텝 2 : 점 4가 속하여 있는 선의 끝점 1과 9를 찾는다.

스텝 3 : 점 1이 시작점이 되고 이 점 1에 8 방향 벡터를 차례로 적용하여 점 2를 찾는다. (방향벡터 4)

$$(x_1, y_1) + V_1 = \text{out of counter}$$

$$(x_1, y_1) + V_2 = \text{out of counter}$$

$$(x_1, y_1) + V_3 = \text{out of counter}$$

$$(x_1, y_1) + V_4 = \text{counter}$$

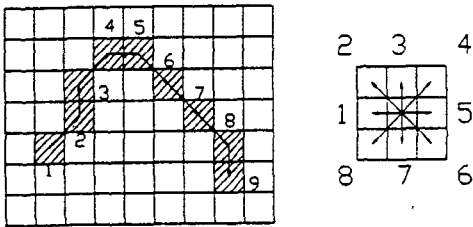


Fig. 13 Finding the contour and direction of vector

스텝 4 : 방향벡터 4와 반대방향의 벡터 8의 다음 벡터인 1부터 점 2에 적용하여 점 3을 찾는다.

$$(x_2, y_2) + V_1 = \text{out of counter}$$

$$(x_2, y_2) + V_2 = \text{out of counter}$$

$$(x_2, y_2) + V_3 = \text{counter}$$

스텝 5 : 이와 같이 계속하여 끝점 9가 찾아지면 끝낸다.

Fig. 14에 전체적인 윤곽생성처리과정이 나타나 있다.

### 3. 결 론

본 연구를 통하여 얼굴화상의 해석, 변환, 특징선의 추출 및 드로잉규칙의 설정에 관한 종합적인 고찰이 있었으며 이러한 방법은 사람의 얼굴을 대상으로 하는 향후의 각종 처리에 응용 될 수 있을 것이다. 그리고 특정한 바와 같이 본 연구에서 사용된 처리방법은 인종을 초월한 다양한 모델에 대하여 무리없이 적용될 수 있을만큼 강인하였으며

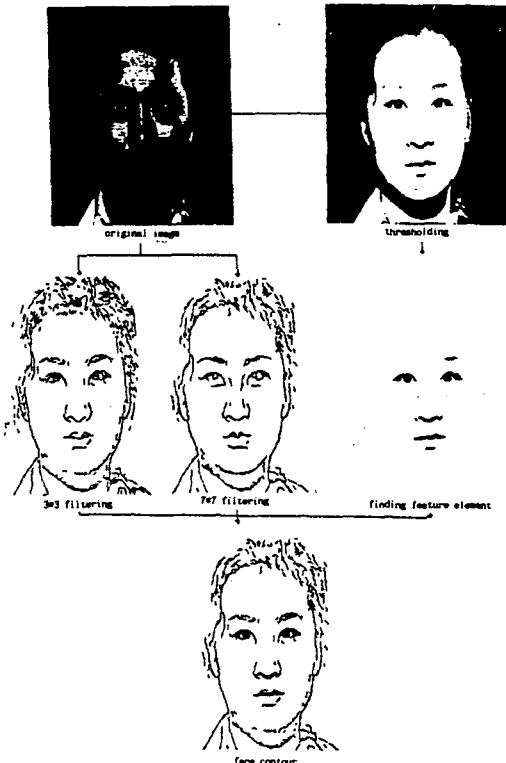
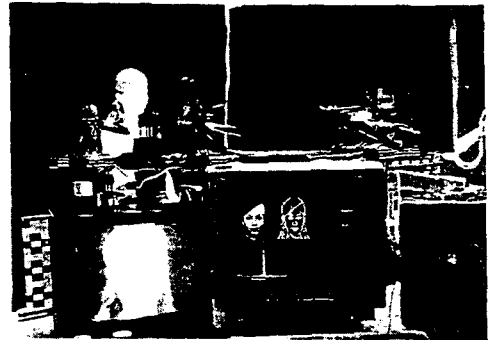
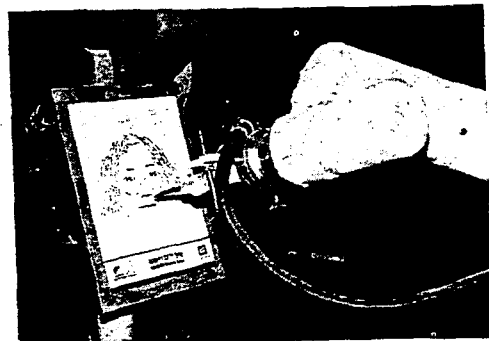


Fig. 14 Example of the contour generation



(a) Image capturing and processing system



(b) Drawing execution using robot

Fig. 15 Whole system



대부분의 경우 처리결과 또한 양호하였다.

### 참고문헌

1. Chang Seok Choi, Toru Okazaki, Hiroshi Harashima and Tsuyoshi Takebe, 1991, "A System of Analyzing and Synthesizing Facial Images," *IEEE International Symposium on Circuits and Systems*, pp. 2665~2668.
- (2) Chang Seok Choi, Hiroshi Harashim and Tsuyosi Takebe, 1991, "Analysis and Synthesis of Facial Expressions in Knowledge-Based Coding of Facial Image Sequence." *International Conference on Acoustics Speech, and Signal Processing*, pp. 2737~2740.
- (3) Hiroji Takano, Daisuke Ohkawara, Akiyoshi Nakada, and Fujio Nakajima, 1985, "Portrait-Drawing Robot System," *National Technical Report*, Vol. 31, No. 4, pp. 1380~1386.
- (4) しおの みつる, 武田 哲, 村山智彦, 1988, "顔画像からの似顔畫作成システム," *Television 학회지*, Vol. 42, No. 12.
- (5) Yasushi Sumi and Yuichi Ohta, 1992, "A Scheme for Human Face Analyzer which can Accept Various Apperances," *전자정보통신학회 논문지 D-2 Vol. J75-D-2 No. 2*, pp. 236~245.