

실사형 캐리커처 생성을 위한 형태 정보 추출 및 음영 합성

박 연 출[†] · 오 해 석^{**}

요 약

본 논문은 윤곽선만을 추출하여 캐리커처를 생성하던 기존의 시스템과 달리 음영을 윤곽선과 합성하여 캐리커처를 생성하는 캐리커처 생성 시스템을 제안한다. 이 방식을 사용할 경우 얼굴의 텍스처에 정보까지 생성시 고려하기 때문에 좀 더 실사형에 근접한 캐리커처를 생성할 수 있다. 본 논문에서 제안하는 시스템은 벡터를 기반으로 하기 때문에 사이즈에 제한 없이 자유로운 변형이 가능할 뿐만 아니라 2D 캐릭터에 자유로운 표정을 적용하는 데에도 쉽게 적용이 가능하다. 또, 벡터의 특징으로 인해 모바일 상에서도 적은 용량으로 이용 가능하다. 본 논문은 벡터 형태의 캐리커처를 생성하는 방법과 음영을 제작 및 합성하는 방법을 함께 제시한다.

Appearance Information Extraction and Shading for Realistic Caricature Generation

Yeon-Chool Park[†] · Hae-Seok Oh^{**}

ABSTRACT

This paper proposes caricature generation system that uses shading mechanism that extracts textural features of face. Using this method, we can get more realistic caricature. Since this system is vector-based, the generated character's face has no size limit and constraint, so it is available to transform the shape freely and to apply various facial expressions to 2D face. Moreover, owing to the vector file's advantage, it can be used in mobile environment as small file size. This paper presents methods that generate vector-based face, create shade and synthesize the shade with the vector face.

키워드 : 얼굴 검출(Face Detection), 얼굴 특징추출(Facial Feature Extraction), 캐리커처 생성(Facial Caricaturing)

1. 서 론

인터넷이 보편화 되고, 콘텐츠 사업의 비중이 증대됨에 따라 가상공간에서 나를 대신하는 아바타(Avatar) 및 캐릭터 사업이 점차 큰 부가가치 사업으로 여겨지고 있다. 최근에는 캐릭터에 감정을 부여하여 감정 에이전트(Emotional Agent)로도 이용하고, 온라인상에서 제작된 캐릭터를 수천 가지가 넘는 기존의 오프라인 캐릭터 상품으로까지 제작, 그 영역을 확장하여 그 활용 분야도 다양해지고 있는 추세이다.

자동으로 생성된 캐릭터의 얼굴을 기존 만화형 캐릭터 대신 이용할 경우 마치 그 캐릭터가 실제 본인과 같은 일체감을 얻을 수 있고, 반면 사진과 달리 실제 얼굴이 공개되지 않으므로 익명성도 보장하여 인터넷에 적합한 대체

ID로 적절히 이용할 수 있다는 장점을 갖는다.

그러나 이러한 장점에도 불구하고 캐릭터 제작 과정이 전문 디자이너에 의해 수작업으로 이루어지고 있어 많은 인력과 시간 낭비를 초래하고 있다.

본 논문은 이렇게 수작업으로 이루어지고 있는 캐릭터 생성 작업을 자동화하기 위해 필요한 각 이목구비별 특징 정보 및 특징정보 추출 방법, 그리고 각 구성요소별 분류기준에 관하여 언급한다.

얼굴의 특징을 분석하여 이를 이미지화하는 연구는 얼굴 인식에 비하여서는 많이 진행되어져 오지 않았으나 대표적인 연구로는 다음과 같은 것들이 있다.

캐리커처 생성에 관한 연구는 Brennan [1]이 맨 처음으로 캐리커처 생성기(Caricature Generator)를 발표하면서 시작되었다. 후쿠이 대학(Fukui University)의 캐리커처 생성 시스템은 사람 생각의 애매함과 복잡함을 단순화시켜 처리하는 퍼지셋(fuzzy-set)을 이용하여 캐리커처를 생성하는 시스템이다[2, 6-8]. 표준 얼굴 이미지의 21개의 특징(눈의

† 준 회원 : 숭실대학교 대학원 컴퓨터학과
** 종신회원 : 경원대학교 소프트웨어대학 교수
논문접수 : 2004년 1월 28일, 심사완료 : 2004년 4월 26일

크기, 눈썹두께, 코의 위치, 높이, 얼굴형태, 머리 길이 등)에 파라미터를 이용하여 입력된 이미지의 특징 정보를 추출한다. 이렇게 추출된 각각의 크기를 49개의 특징 단어 (big, small, round, thin, tall, short 등)를 이용하여 단어들의 분석에 따라 선택하여 얼굴에 배치시킴으로서 캐리커처를 생성한다.

츠크 대학(Chukyo University)의 PICASSO System은 생성된 캐리커처의 감정 표현을 목적으로 개발된 시스템이다. 이 시스템의 캐리커처 생성 방법은 표준 얼굴 이미지의 윤곽선을 455개의 포인트로 표현하고 입력된 이미지에 455개의 포인트를 이용하여 각각의 포인트를 비교하여 그 포인트를 확대와 축소로 조절하여 포인트를 선으로 이어 윤곽선을 그려줌으로써 캐리커처를 생성한다[3, 10].

Hong Chen[12]은 비모수적 샘플링 기법을 이용한 얼굴 스케치 생성방식을 제안하였다. 제안한 방식은 학습과 비모수적 샘플링 기법을 이용해서 스케치를 자동으로 생성해준다. 예술가에 의해 그려진 스케치 이미지들을 DB화 해놓고, 그 DB의 샘플링 이미지들을 변형하여 합성하는 방식을 사용하였다.

그러나 이러한 기존 연구들은 다음과 같은 문제점을 안고 있다. 첫째로 다양하지 않은 특징정보를 이용하여 생성하므로 얼굴의 다양성이 반영되지 않는다는 것이며 둘째로 표준 얼굴 이미지를 기반으로 변형작업을 통하여 캐리커처를 생성하거나 윤곽선(edge)만을 이용하여 얼굴의 텍스처어(texture)는 고려되지 않은, 입력된 윤곽선만을 변형, 생성하는 방식이라는 것이다. 그리고 모두 컬러 방식이 아닌 흑백 방식이다. 이는 텍스처를 고려하지 않았기 때문인데 현재 대부분의 아바타 및 캐릭터 그리고 어플리케이션들이 컬러기반인 상황에서 하나의 단점으로 여겨진다.

본 논문은 이러한 문제점을 해결하기 위해 몽타주 기법과 음영합성 기법을 이용한, 디자이너(예술가)의 감각을 살린 벡터기반의 캐리커처 생성 시스템을 제안한다. 본 논문은 이목구비의 모양 특징정보와 상관계수를 이용한 이목구비 이미지의 검색기법에 대해 제안하였으며, 검색속도의 향상과 검색 오류를 줄이기 위해 사용된 각 이목구비 이미지들의 카테고리 분류 기준을 제시하였다(검색대상을 줄임). 또, 기존 연구에서처럼 윤곽선만으로는 그 사람의 느낌을 살리는데 한계가 있다고 판단하여 사진으로부터 음영을 추출하여 이를 음영판으로 만들고 검색된 이목구비 이미지와 합성하는 방식을 제안하였다. 여기서 사용된 이목구비 이미지는 실제로 디자이너에 의해 그려진 것들로써 디자이너의 감각이 결과물에 반영되도록 하였다.

원래 캐리커처란 주로 정치적인 만평이나 특정인의 얼굴의 특징을 과장해서 그럴 때 쓰인다[1]. 하지만, 본 연구는 얼굴의 과장보다는 닮은 얼굴을 생성해 내는데 초점을 두고 진행되어왔다.

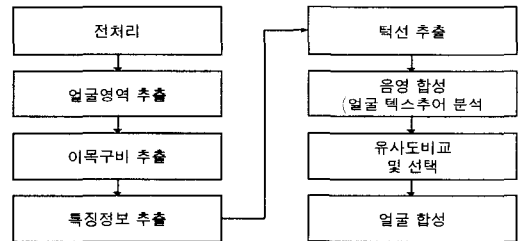
본 논문의 구성은 다음과 같다. 2장에서는 캐리커처 생성

시스템의 구조에 대해 설명하고, 3장에서부터 6장까지는 각 모듈별 기능 및 연구 성과에 대해 설명한다. 7장에서는 음영합성 과정에 대해 구체적으로 설명한다. 8장에서는 검색된 이목구비를 얼굴형위에 매핑시키는 과정에 대해 설명한다. 9장에서는 이 논문의 실험결과 및 분석에 대해 논하고 10장에서 결론을 맺는다.

2. 캐리커처 생성 시스템 구조

본 논문은 크게 입력된 정면의 사진에서 특징정보를 추출하는 과정과 데이터베이스에서 유사한 이목구비 이미지를 추출하여 매핑시키는 과정으로 나누어진다. 구체적으로 설명하면 다음과 같다.

첫째, 정면의 입력된 인물 사진에서 얼굴 영역을 추출한다. 둘째, 얼굴영역에서 수직, 수평 히스토그램을 이용하여 이목구비(눈, 코, 입, 눈썹) 등을 추출한다. 셋째, 추출된 이목구비에서 위치, 크기, 모양 등 다양한 특징정보를 추출한다. 넷째, 사진으로부터 음영정보를 추출해서 음영판을 만들어낸다. 다섯째, 특징정보를 이용하여 데이터베이스에 저장되어 있는 이목구비 이미지와 추출된 이미지의 유사도를 측정한다. 여섯째, 가장 유사한 이목구비 이미지를 검색하여 만들어진 벡터 얼굴형에 각각의 눈, 코, 입, 눈썹 등을 매핑시킨다. (그림 1)은 2D 얼굴 생성 시스템의 구조와 데이터 흐름을 함께 표현한 것이다.



(그림 1) 시스템 구성도

입력된 이미지에 캐릭터 생성 과정을 간략하게 설명하면 다음과 같다.

- ① 영역 추출
 - 얼굴 영역 추출 : 명도와 색상 정보 이용
 - 이목구비 추출 : 히스토그램 이용
- ② 특징정보 추출
 - 이목구비 특징 추출
 - 특선 특징 추출, Bezier Curve 형태로 윤곽선 추출
- ③ 음영 합성
 - 사진으로부터 음영 등고선 추출(텍스처어 분석)
 - 각 단계별 색 적용
- ④ 유사도 계산
 - 동일 클래스 찾기

- DB에 분류된 이목구비의 특징정보와 유사도 계산
 - 유사한 이목구비 이미지 검색
- ⑤ 얼굴 합성(음영판위에 이목구비 매핑)
- 추출된 특징정보 위치에 이목구비 매핑

본 시스템의 입력 이미지는 정면을 보고 있는 상반신 사진이다. 입력된 이미지에서 얼굴 영역을 추출하고, 이어서 이목구비(눈, 코, 입, 눈썹)의 부분영역을 추출하고, 추출된 이목구비로부터 특징정보를 추출하고, 미리 구축해 놓은 이목구비 특징정보 데이터베이스를 이용하여 각각의 영역에서 가장 유사도가 높은 이미지를 검색해 낸다. 본 논문에서 이용하는 이목구비 데이터베이스는 전문 디자이너에 의해 그려진 이목구비를 각각의 특징에 맞게 특징정보와 함께 분류해 놓은 것이다. 추출된 각 이목구비를 얻어진 특징정보에 의해 매핑시켜, 사용자에게 보여줄 수 있도록 설계되었다.

3. 2단계 얼굴영역 추출

얼굴 영역을 추출하기 위해서는 사람의 얼굴이 갖는 컬러의 분포 형태를 가정하는 것이 우선 필요하다. 이에 본 논문에서는 약 200장의 사진으로부터 HSI 컬러 스페이스의 벡터값(h, s)을 얻어내어 이를 수치화 하는 작업을 수행하였다.

지금까지는 컬러 차트를 이용하여 컬러값을 얻어내는 방식을 사용하였으나 이 방식은 시간과 계산량이 많이 요구되는 단점을 가지고 있다. 이에 본 논문은 이전의 많은 논문이 사람의 얼굴이 갖는 컬러가 정규분포를 띤다고 가정하였으므로[10, 11, 18-20] 굳이 컬러 차트를 사용할 필요가 없다는데 착안하여 평균과 분산(표준편차)만으로 사람의 얼굴의 분포를 수치화하여 사용하였다. 이는 기존의 논문이 얼굴의 영역을 찾아낼 때 유사한 얼굴 영역을 찾아내는 과정에만 컬러 차트를 사용하였는데 이는 계산량이 증가하여 결국에는 전체적인 분석시간의 증가만 초래하였다. 하지만, 본 논문이 제안하는 방식은 1단계 추출에서는 컬러 분포를 정규분포라 가정하고, 평균과 표준편차를 이용하여 신뢰구간내의 유사 얼굴 영역을 추출하고, 그 영역내에서 일단 눈동자를 찾은 후, 단일 사진내에서 추출된 유사 얼굴 영역내에서 다시 한번 샘플링을 하여 그것을 차트화 하고 이를 통해 정확한 얼굴 영역을 얻어내는 2단계 방식을 사용하였다. 이와 같은 방법을 사용하는 이유는 사람의 얼굴이 갖는 얼굴 컬러의 분포는 어느 정도 유사하지만 실제로 각 사진마다 갖는 얼굴의 분포는 사진별로 조금씩 차이를 보이기 때문이다. 이 차이는 각 사진마다 조명, 화장, 카메라의 종류, 찍는 사람의 능력에 따라 다르게 나타난다. 즉, 각 사진이 갖는 특성을 그대로 적용하여 얼굴을 추출하기 위해서는 그 사진 내에서 직접 정보를 얻어내어야 한다는 결론을 얻을 수 있다. 이에 본 논문은 1차적으로 통계적 모멘트 값

을 이용한 얼굴 영역의 검출 후, 2차적으로 다시 사진 내에서의 얼굴의 컬러 분포의 모멘트 값을 얻어내어 얼굴의 영역 및 윤곽을 추출해 내는 2단계 컬러 샘플링 방식을 사용하였다.

본 논문에서 HSI 컬러 스페이스를 사용한 이유는 알려진 바와 같이 이 방식이 사람 시각의 컬러 인지 방식과 가장 흡사하기 때문이다. 얼굴 컬러의 추출은 HSI 값 중에서 HS 값만을 벡터 형태로 취하였다. N이 샘플의 총 개수라고 할 때, 샘플 하나의 값은 $(h_i, s_i), i = 1, \dots, N$ 이다. 여기서 명암 값(Intensity)을 사용하지 않은 이유는 얼굴에서 명암값의 분포가 균일하지 못하고 얼굴 전체에 걸쳐 다양하게 퍼져있기 때문이다. 이는 조명이나 화장에 의해 영향을 많이 받기 때문이기도 하다. 따라서, HS값만을 사용하게 되면 조명이나 화장에 강한 장점을 갖는다.

얼굴색의 분포 모델을 얻기 위해 사용된 사진의 개수는 총 194장(남 : 97, 여 : 97)이며, 사진의 획득은 인터넷을 통해 이루어졌다. 모두 임의의 유저로부터 직접 제출받은 것들이다. 따라서 랜덤하게 추출되어졌다고 볼 수 있다. 단, 가로, 세로 100픽셀 이상이고, 정면이라는 전제가 있었다. 다음으로 사진을 비율에 맞추어 가로 최대 290, 세로 최대 390 사이즈로 정규화 하였다.



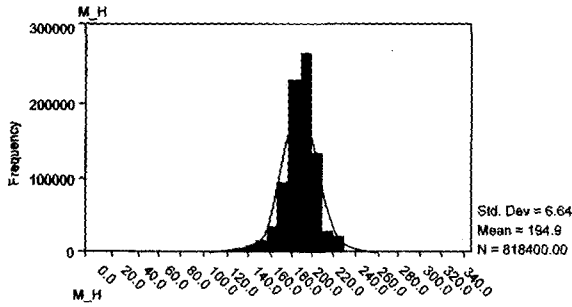
(그림 2) 살색 표본 추출

샘플링은 (그림 2)에서 보는 바와 같이 가능하면 얼굴 영역 전체에 걸쳐서 골고루 추출될 수 있도록 하기 위해, 사진에서 얼굴 부분 중 14부분을 선택하여 직사각형(20×30) 형태로 추출하였다. 600픽셀, 총 $194 \times 14 \times 600 = 1,629,600$ 픽셀. 이 값들을 이용해 $E(H), E(S), Var(H), Var(S)$ 를 구한 후, 이를 이용해 신뢰구간을 획득하였다.

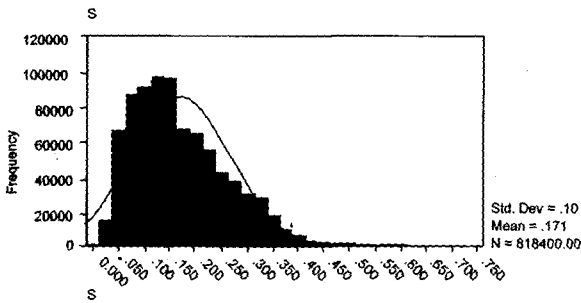
신뢰구간은 $\bar{x} - 3\sigma < \mu < \bar{x} + 3\sigma$ 이다. 여기서, α 값을 0.01로 했을 경우, 신뢰수준은 99%이고, 오차범위는 0.5%이다.

(그림 2)는 사진으로부터 살색을 샘플링 하는 과정을 표현한 것이다. 총 14개의 영역으로부터 샘플링이 이루어졌다. 이 14개의 영역의 선택은 얼굴 전체를 골고루 대표할 수 있고, 변화가 심하지 않은 곳이어야 한다는 기준에 의해 이루어졌다. 피부가 아닌 샘플이 추출될 가능성을 줄이기 위하여 자동 추출을 하지 않고 수작업으로 행하였다.

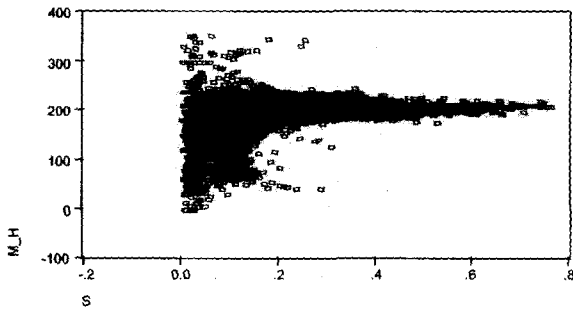
(그림 3)과 (그림 4)는 얻어진 데이터를 히스토그램 형태로 표현한 것이다. 그림 안의 곡선은 정규분포를 표시한 것이다. 또, (그림 5)는 Hue와 Saturation값을 정규화하여 2차원 공간에 산포도로 표현한 것이다.



(그림 3) 변형된 Hue값의 히스토그램



(그림 4) Saturation값의 히스토그램



(그림 5) Hue와 Saturation 산포도

4. 얼굴 특징정보 추출

4.1 눈동자 추출

눈동자의 추출에 앞서 이목구비 영역의 추출이 선행되어야 하는데 얼굴의 대칭성을 이용한 명암의 수평, 수직 히스토그램[4] 분석과 한국인의 평균 얼굴형 모델을 통하여 얻어질 수 있다[14]. 즉, 히스토그램을 이용하여 이목구비의 위치를 예측해 낸 후, 이 위치의 적합성의 판단을 위해 평균 얼굴형 모델의 수치가 사용되어졌다. 여기서 얻어진 눈의 부분영역이 눈동자의 추출시 사용되어지는 입력 영상으로 사용되어진다. 눈동자의 추출은 얼굴인식 분야에서 중요한 파트중의 하나이다. 왜냐하면, 사진에서 얻어낼 수 있는

가장 신뢰할 만한 특징점일 뿐만 아니라 눈동자 사이의 거리가 여러장의 사진들을 정규화하는데 가장 기본이 되는 기준치가 되기 때문이고, 또한 사진 속 얼굴의 수평을 맞추는데도 사용되기 때문이다.

눈동자를 찾기 위해 가변 템플릿 수렴방식을 제안한다. 템플릿은 크기가 가변적인 정원을 사용하였다. 눈동자가 위치할 만한 후보 영역을 우선 찾아내어야 하는데 이는 명암값의 수직 수평 프로젝션을 통해 얻어낼 수 있었다. 이렇게 얻어진 좌측 눈 영역과 우측 눈 영역이 각각 템플릿 매칭이 이루어지는 영역으로 사용된다.

이렇게 얻어진 영역에 전처리 과정으로 눈동자에 반사된 빛을 제거하는 과정이 필요하다. 식 (1)은 3×3 마스크의 최소값을 이용하여 반사된 빛을 제거한다.

$$f'(x, y) = \min_{i=-d, d, j=-d, d} f(x+i, y+j) \quad (1)$$

식 (2)~식 (6)은 눈동자를 찾기 위해 사용된 템플릿과 최적해를 구하는 수식이다.

$$x^2 + y^2 \leq r^2, \text{ where } \begin{aligned} x &= 1, 2, \dots, N1, \\ y &= 1, 2, \dots, N2, \\ r &= 15, 14, \dots, n \end{aligned} \quad (2)$$

$$S(w_{p,r}) = \frac{1}{N_r} \times \left(\sum_x \sum_y ((255 - I_{x,y}) * W) \right) \quad (3)$$

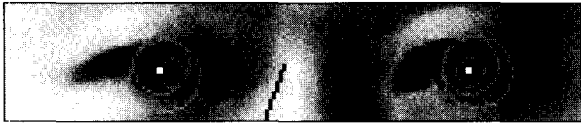
$$V(w_r) = \frac{1}{N_r} \times \sum_{i=1}^{N_r} (I_i - \bar{I})^2 \quad (4)$$

$$w_r = \operatorname{argmax}_w S(w_{p,r}) \quad (5)$$

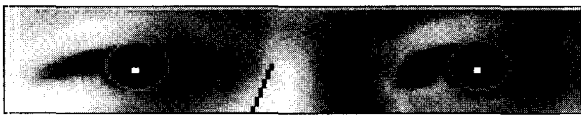
$$w_{opt} = \operatorname{argmax}_w D(V(w_r), V(w_{r-1})) \quad (6)$$

여기서, x, y는 각각 이미지의 중심으로부터의 수평, 수직 인덱스이며, r은 반지름을 의미한다. 이렇게 얻어진 템플릿의 특징 벡터(X, Y, R, S, V) 값을 이용하여 최적의 후보를 찾아내게 된다. 식 (3)에서 S는 반지름이 같은 템플릿 내에서의 최적의 후보를 찾는 데 사용되는 밝기 값의 평균을 가리키며 W는 가중치로써 중심에 가까울수록 밝기 값이 어두울수록 S값을 크게 하는 역할을 한다. $w_{p,r}$ 는 중심 좌표가 p이고 반지름이 r인 템플릿(윈도우)을 가리키며, w_r 는 반지름이 r일 때 S값을 최대로 하는 윈도우를 가리킨다. 식 (4)에서 V는 반지름이 서로 다른 템플릿 중에서 최적의 후보를 찾기 위해 사용되는 모멘트 값이다. V값으로 본 논문에서는 2차 모멘트(분산)를 사용하였다. 분산을 사용한 이유는 눈은 하얀색 흰자와 검은색 눈동자로 구성되어지기 때문에 최적의 눈동자의 크기를 얻었다고 했을 때 명암의 변화율이 가장 낮기 때문이다. 또, 단순히 같은 반지름을 갖는 템플릿 중에서 분산의 값이 가장 작은 템플릿을 선택하는 것이 아니라, 반지름이 r-1일 때의 분산과 r일 때의 분산의 차가 가장 클 때의 원을 최적의 해로 보았다. w_{opt} 는 눈동자를 가장 잘 포함하는 최적해 윈도우를 가리

킨다. 이렇게 두 번의 비교 과정을 거침으로써 명암 이미지에서 검은 눈동자를 가장 최적으로 포함하는 눈동자의 후보를 찾아낼 수가 있다. (그림 6)은 템플릿의 반지름을 수렴시켜가면서 S값을 비교, 값이 가장 큰 지점을 표시한 것이다.

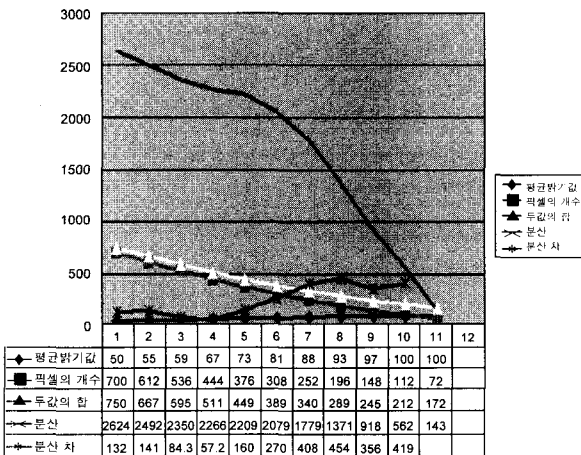


(그림 6) 탐색 과정(템플릿 사이즈의 수렴)



(그림 7) 추출된 눈동자의 위치와 크기

(그림 7)은 각 반지름별 최적 후보끼리 분산차를 구하여 분산차가 가장 큰 최종 최적해를 표시하고 있다. 그림처럼 눈동자의 위치와 크기를 동시에 얻어낼 수 있다. (그림 8)은 템플릿의 각 반지름별 분산 및 반지름간의 분산차를 그래프 형태로 표현한 것이다. 보이는 바와 같이 그래프 상에서 분산차가 최대가 되는 지점이 최적해로 선택되어 진다. 이렇게 얻어진 눈의 좌표를 이용하여 이미지를 다시 회전시킨다. 회전을 시키는 이유는 얼굴의 수평을 맞추기 위해서이다. 눈동자가 얼굴 중에서 가장 찾기 쉬운 객체이고, 오인할 확률도 적기 때문에 눈동자가 항상 얼굴의 가장 기본적인 기준점이 된다.

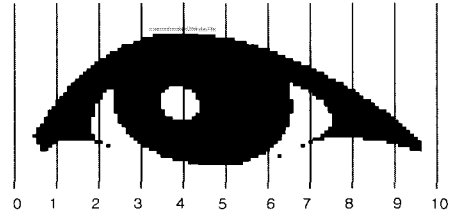


(그림 8) 분산과 분산차의 그래프

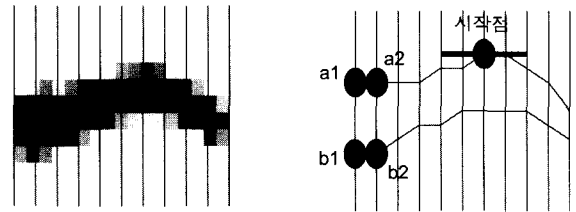
4.2 이목구비의 특징정보 추출

추출한 이목구비를 이진화한 후 유사도 비교를 행한다. 이때, 이목구비 영상은 매우 작아서 약간의 잡음(noise)에도

민감하기 때문에 잡음제거가 우선적으로 필수적이다. 이목구비 영상 데이터의 에지를 흐리게 하지 않고 잡음을 제거할 수 있도록 가장 대표적으로 많이 쓰이는 미디언(Median) 필터를 전처리 과정에서 사용하였다. 전처리가 끝난 후에는 이진화 되어있는 이목구비 영상에 대한 특징정보를 추출한다.



(그림 9) 윤곽선과 구간별 눈의 특징정보 추출



(그림 10) 왼쪽 눈썹의 특징정보 추출

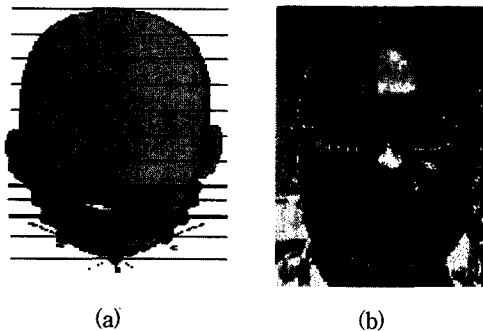
(그림 10)은 이목구비 중 왼쪽 눈썹의 특징정보를 추출하기 위한 한 예이다. 우리는 그림처럼 왼쪽 눈썹을 가로로 10등분을 하고, 10등분된 선에 대한 각각의 시작점과 끝점, 그리고 한 점에서 다음 점까지((a1, a2)···, (b1, b2)···)의 기울기와 10개의 눈썹 높이 등을 특징 벡터 형태로 표현, 눈썹의 특징 정보를 얻어낸다. 이때 사용되는 수치는 모두 가로 폭으로 정규화된 값이다. 이와 같이 눈과 입술도 똑같은 형태로 특징추출이 행해진다. 입술 같은 경우는 이에 덧붙여 윗입술의 두께, 아랫입술의 두께 등도 특징정보로 추출되어진다.

4.3 턱선 특징정보 추출 및 벡터화

턱선의 특징정보는 우선 턱선의 형태를 벡터로 표현한 후, 이들 벡터 데이터를 표준 얼굴형과 비교하여 얻어낼 수 있다. 본 논문에서는 턱선의 형태를 벡터로 추출하기 위해 국소영역분할 방식을 이용하여 턱선의 특징점들을 추출해낸다. 이때 특징점들의 추출 오차를 줄이기 위하여 얼굴 내에서의 살색을 샘플링한 후에 그 데이터를 기초로 다시 세그먼테이션을 행하는 방식을 취하였다.

턱선의 특징점 추출은 음영차(윤곽선)만을 이용하지 않고, 색상도 함께 이용하여 얼굴 경계 추출의 정확도를 높였다. 만약 음영만을 이용하였다면 살색과 유사한 헤어를 갖는 사진일 경우 헤어와 얼굴을 구분해 내지 못할 것이다. 물론 그 역의 경우도 발생 가능하다. 여기서 중요한 문제로 대두되는 것이 턱선의 특징점 후보들 중에서 최적해를 어떻게 찾아내느냐 하는 것이다. 즉, 음영과 컬러값들을 이용

하여 세그멘테이션을 행할 경우 각각의 방법으로 얻어진 특징점 후보들이 항상 일치하지는 않는다. 따라서, 본 논문에서는 각 추출된 특징점 후보들과 표준 얼굴 모델의 특징점들과의 euclidean distance값을 구하여 거리차가 가장 적은 특징점을 최적해로 사용하였다. 이때 사용된 표준 얼굴 모델의 특징점들은 두 눈동자 사이의 거리값에 의해 유추해낸 양 눈의 연장선이 만나는 얼굴의 끝점과 아래 턱점, 그리고, 윗입술의 상단, 입술의 중앙, 입술의 하단 점 등이다. 이렇게 얻어진 얼굴의 13개 특징점들을 Bezier Curve로 연결하여 얼굴형을 만들어낼 수 있었다(그림 11). 만들어진 Bezier Curve는 귀 이미지와 함께 벡터 파일(*.EMF)로 저장된다.



(그림 11) (a) 턱의 특징점 (b) 턱선 및 이목구비별 형태정보 추출

5. 이미지 분류(Classification)

일반적으로 이미지 분류란 검색 대상이 되는 이미지를 분류하여 검색되는 후보의 수를 줄이기 위하여, 물리적 대상이나 사건을 의미 있는 특징들의 추출을 기반으로 사전에 정의된 클래스 중 하나로 할당하는 것을 말한다. 본 논문에서 우리는 속도향상을 위해서 이러한 개념에 근거하여 하나의 이미지로부터 추출된 다수의 특징정보에 따라 이목구비 이미지를 특정 클래스로 분류해 낸다. 이 분류의 기준은 위 4장에서 설명한 각 이목구비의 특징정보를 이용한다. 그리고 본 논문에서 클래스화된 눈, 코, 입, 얼굴형, 눈썹들의 수는 <표 1>과 같이 각 부분별로 해당하는 수만큼 데이터베이스에 존재하고 있다. 입력된 얼굴 이미지는 이목구비의 특징정보를 이용하여 나누어진 다수의 클래스에서 가장 유사도가 높은 것을 추출하여 이목구비를 매핑시키게 된다. 좀 더 구체적인 각 이목구비별 분류방법은 기존 논문에서 언급한 바 있다[17].

<표 1> 벡터 이미지 개수

종 류	수 량
눈 썹	각 200
눈	각 200
코	20
입	20

6. 검색 및 유사도 계산

일반적으로 이미지 검색은 유사 이미지를 검색하는 목적에 따라 색상 정보만을 적용할 때도 있고, 모양정보만을 적용할 때가 있다. 또는 두 특징정보의 적절한 배합으로 검색하고자할 때가 있을 수 있다.

입력된 이미지에서 수직, 수평 히스토그램을 이용하여 추출되어진 이목구비 이미지의 색상은 다르고, 모양은 거의 같다. 그리하여 본 논문에서는 위와 같이 모양이나 색상을 이용한 특징정보를 이용할 수 없다. 그래서 본 논문에서는 특징정보 데이터와 이목구비 이미지와의 유사도를 계산하여 이미지 검색을 시도하였다. 눈과 눈썹 이목구비 윤곽선의 유사도를 계산하기 위해 사용한 유사도 공식은 다음과 같다.

$$S = r \times \omega + (1 - |\alpha - \beta|) \tag{7}$$

$$r = \frac{\sum(A_i - E_A)(B_i - E_B)}{\sqrt{\sum(A_i - E_A)^2 \sum(B_i - E_B)^2}} \tag{8}$$

여기서, 식 (8)은 상관계수이다. 이 식에서 A_i 는 입력된 이미지에서 추출된 이목구비인 A의 구간별 높이를 의미하고 B_i 는 데이터베이스에 있는 이미지인 B의 높이를 의미하며, E_A, E_B 는 A와 B의 평균 높이를 의미한다. 상관계수인 r의 성질은 ±1에 가까울수록 강한 상관관계를 갖는다. 하지만, 본 논문에서는 음의 상관관계는 의미가 없으므로 고려하지 않는다. 그러므로 r의 값은 0부터 1의 양의 실수 값을 갖는다. 또한 식 (7)의 α, β 는 A_i, B_i 각각의 가장 큰 가로와 세로의 비를 의미한다. 그리고 ω 는 가중치이다.

식 (7)의 값이 클수록 윤곽선 형태의 유사성이 높음을 의미한다.

7. 음영 합성(Shading)

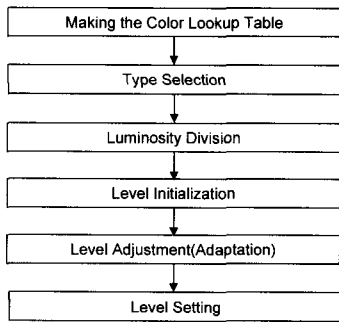
음영판 제작은 캐릭터 얼굴 제작에 있어서 또 다른 새로운 이슈거리 중 하나이다. 왜냐하면 눈, 코, 입, 얼굴형만 가지고는 사용자와 닮은 캐릭터를 만들 수 없기 때문이다. 음영이 들어가지 않은 얼굴은 입체감이 없을 뿐만 아니라 그 사람의 이목구비 모양 이외의 특징은 전혀 반영하지 못한다는 단점을 가지고 있다. 이에 본 논문은 이러한 문제를 해결하기 위하여 음영을 사진변형을 통해서 얻어내어 이목구비와 합성하는 방법을 제안한다. 이에 필요한 제작 과정은 (그림 12)와 같다.

사진으로부터 얻어낸 음영을 그대로 사용할 경우 너무 실제 피부색에 가까워 캐릭터의 얼굴로 사용하기에는 적합하지 않다. 즉, 컬러의 부조화가 발생한다. 따라서 디자이너가 흔히 일러스트시 사용하는 컬러를 사용할 필요가 있다. 이에 본 논문에서는 웹상의 캐릭터 얼굴에서 주로 사용되는

컬러를 추출하여 Look-Up Table(LUT) 형태로 구성하였다. 이 때 테이블을 또한 세 종류(Normal, Light, Dark)로 나누어서 구성하였다. Normal 테이블은 일반적으로 가장 사람들이 선호하고 일반적으로 사용되는 컬러로 구성되어 있고, Light 테이블은 약간 밝은 톤으로 구성되어 있어서 사용자가 '뽀샤시'한 얼굴형을 원할 때 선택될 수 있다. 마지막으로, Dark 테이블은 어두운 색 계통의 살색을 모아놓은 것으로서 흑인이나 태닝한 얼굴색을 원할 때 선택될 수 있다.

웹상에서 쓰이는 캐릭터의 얼굴은 실사보다 현실성이 떨어지기 때문에 컬러가 많이 쓰이지 않는다. 따라서, LUT는 많은 컬러로 구성하지 않고, (총구간의 수/2)개의 컬러로 구성하였다. 여기서 총구간의 수는 얼굴영역을 밝기 값에 따라 나눈 구간의 총 개수이다. 실험결과 이 정도의 컬러면 실사형과 만화형의 느낌을 공히 느낄 수 있도록 할 수 있었다. 물론, 경우에 따라 좀 더 실사에 가까워야 할 경우에는 컬러의 수를 증가시키고, 좀 더 만화형에 가까워야 할 경우에는 컬러의 수를 줄임으로써 필요한 결과물을 얻어낼 수 있다.

음영을 사진으로부터 얻어내는 과정은 먼저 얻어진 얼굴 영역을 밝기 값에 따라 14단계로 나누는 과정부터 시작된다. 이것은 히스토그램으로 프로젝션 한 후, 등간격으로 14구간을 나누는 과정이다. 이때 구간의 시작과 끝은 음영의 최소/최대값이 된다. 여기서 14라는 값은 경험치로써 이 값에서 가장 만족스러운 결과를 얻어낼 수 있었다.



(그림 12) 음영판 제작 과정

다음 과정은 얻어진 구간에 대해 해당 구간에 해당하는 컬러를 LUT로부터 참조하여 각 구간 픽셀에 할당하는 과정이다. 이 과정이 끝나면 초기화된 각 레벨을 조정하는 단계로 이어지는데 이 과정의 알고리즘은 (그림 13)과 같다. 밝기값의 조정과정시 기준이 되는 정보는 히스토그램에서 최대 빈도수를 갖는 구간이 된다. 이 구간이 주어진 임계값을 벗어날 경우 레벨 조정이 이루어지게 된다. 즉, 너무 어둡거나 너무 밝으면 이는 웹에서 쓰이는 캐릭터 얼굴의 색상이라고 보기 어렵기 때문에 전체 레벨을 이동시키거나 주변 레벨로 합치게 된다. 알고리즘에서 1번 과정은 전체 레벨을 이동시키는 과정이고, 2번 과정은 각 픽셀별로 레벨을 조정하여 필요하다면 주변 레벨과 합치는 과정을 보이

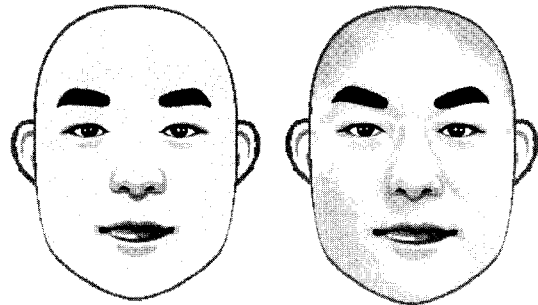
고 있다. 여기서, thn 이 임계값이고, w 는 가중치에 해당된다. 이 두 개의 인자에 의해 레벨이 조정되는데 어떤 값을 지정하느냐에 따라 느낌이 달라지게 되어있다.

```

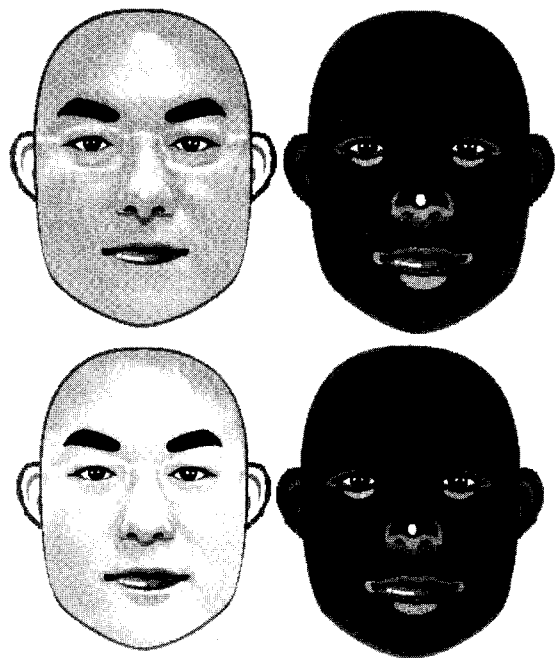
    for every level  $L_i$  of Color Lookup Table
       $Sum = Sum + Gray[TotalLevel - L_i]$ ;
      if (  $Sum/M < th$  )  $Level = Level + w1$ ;
      else if (  $Sum/M > th2$  )
         $Level = Level + w2$ ;
        break;
      else break;

    for every pixel  $A_{ij}$  of the image A
      if ( (  $RegLevel - MaxLevel$  ) >  $th3$  )
        if (  $Percent < th4$  )
           $TableIndex = TableIndex + w3$ ;
        else if (  $Percent < th5$  )
           $TableIndex = TableIndex + w4$ ;
        else  $TableIndex = TableIndex + w5$ ;
         $TableIndex = TableIndex + Level$ ;
         $A_{ij} = Table[TableIndex]$ ;
    
```

(그림 13) 레벨 조정 알고리즘



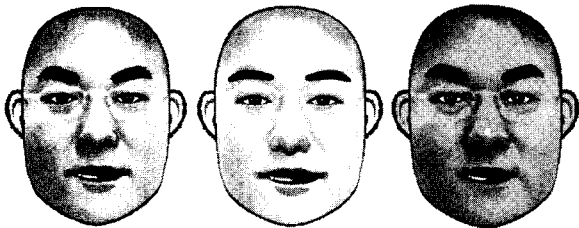
(그림 14) 음영판 없이 합성된 모습과 합성 후의 모습 비교



(그림 15) 레벨 조정 전후 모습

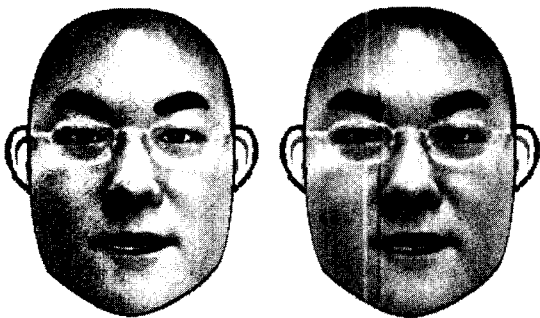
(그림 14)는 음영을 넣기 전과 후의 모습을 비교해 놓은 것이다. 보는 바와 같이 같은 이목구비인데도 음영을 넣는 것과 안 넣는 것의 차이는 굉장히 크다. 왼쪽 얼굴은 사진과 비교하지 않는 한 그 사람의 캐릭터라고 믿기에는 다소 문제가 있다. 반면 음영을 넣은 후의 모습은 그 사람의 캐릭터 얼굴이라고 판단하기에 충분하다.

(그림 15)는 음영 삽입 과정에서 행해지는 레벨의 조정 과정 전후의 모습을 보여주고 있다. 위의 두 그림은 조정 전의 모습이고 아래 두 개의 그림은 조정 후의 모습이다. 우측의 상하 그림은 밝기값의 분포형태를 구분하기 쉽도록 채운 표시 방식의 형태로 표현한 것인데, 그림에서 가장 어두운 부분을 빨간색으로, 가장 밝은 부분을 파란색으로, 그리고 그 중간 밝기는 녹색으로 표현하였다. 즉, 진적색으로 표시된 부분이 가장 밝은 부분이다. 결과적으로, 아래 그림의 빨간색의 비율이 위의 것보다 훨씬 크게 보이므로, 전반적으로 밝은 색이 주류를 이루는 것을 알 수 있다.



(그림 16) Normal, Light, Dark 테이블 적용 후 결과 비교

(그림 16)은 세 가지 종류의 LUT(Normal, Light, Dark)를 각각 다르게 적용한 후의 결과를 보여주고 있다. Light는 전체적으로 밝은 살색이 주류를 이루고 있고, Dark는 전체적으로 어두운 살색이 주류를 이루고 있다. 지금은 이것의 용도가 느낌을 달리하는데 이용되지만 다른 용도로도 쓰일 수가 있다. 즉, 인종을 구분하는데도 이용할 수가 있다. LUT를 Normal, Light, Dark로 구분하지 않고, 황인종, 백인종, 흑인종의 세 가지 형태로 구분하는 것이다. 그렇게 되면, 캐릭터의 국적이 국내에 한정되던 것을 다양한 국적으로 변모시킬 수 있을 것으로 보인다.



(그림 17) 단계별 알파값 적용

(그림 17)은 사진의 음영과 각 이목구비를 합성할 시에 투명도(알파값)를 달리하여 표현한 것이다. 투명도를 조절함으로써 만화형과 실사형의 비율 조절이 가능하다. 즉, 투명도를 낮출수록 만화형에 가까워지고, 투명도를 높일수록 실사형에 가까워진다. 즉, 좀더 사진의 형태에 가까워지는 것이다.

8. 이목구비 매핑

이목구비 매핑은 각각의 이목구비 이미지들을 얻어진 위치특정정보에 따라 적절히 배치시켜 캐리커처의 얼굴을 재구성하는 것을 의미한다. 입력된 얼굴 이미지에서 추출된 이목구비 데이터와 이목구비 데이터베이스에 저장된 각 벡터 이미지들 사이의 유사도를 계산하여 유사도가 가장 높은 각 이목구비를 검색하여 이미지를 매핑시키는 작업이다. 이 시스템에서 매핑시키는 위치는 입력된 데이터에서 추출한다. 즉 입력된 데이터의 이목구비를 추출할 때 이목구비의 위치를 찾아내고 그 위치정보를 이용하여 매핑시에 이용한다. 그러므로 각 개인의 이목구비 위치 특징을 이용할 수 있다.



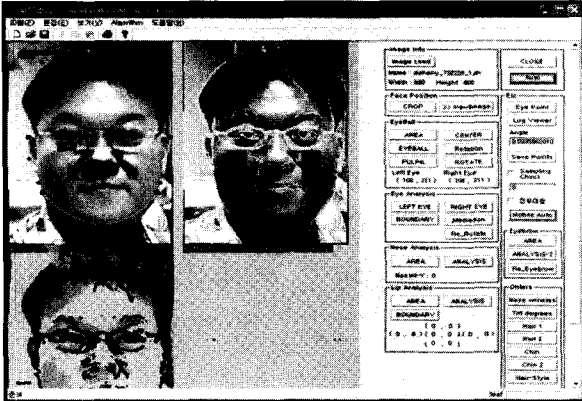
(a) 벡터로 만들어진 얼굴형 (b) 매핑된 얼굴형
(그림 18) 합성얼굴 제작

(그림 18)에서 좌측의 그림은 (그림 11)에서 보여졌던 바와 같이 텍선의 특징점들을 추출한 후 이들 점들을 Bezier Curve로 연결하여 얻어진 얼굴형이다. 또, 우측의 그림은 이 얼굴형에 음영과 추출된 이목구비를 매핑시켜 얻어진 최종 결과물이다. 여기서 귀는 미리 만들어진 벡터 파일로서 매핑시 다른 이목구비와 함께 매핑시킨 것이다.

9. 실험 결과 및 성능 분석

실험에 사용된 사진은 총 194장(남 : 97, 여 : 97)이었으며, 사진의 획득은 모두 인터넷을 통해 이루어졌다. 모두 임의의 유저로부터 직접 제출받은 것들이다. 따라서, 랜덤하게 추출되어졌다고 볼 수 있다. 단, 사진의 크기(가로, 세로 100픽셀 이상)와 정면 사진이라는 전체가 있었다. (그림 19)는 캐리커처 생성 시스템의 전체적인 외관이며 특징 추출이 이루어지고 있는 모습을 보여주고 있다. (그림 20)은 특

징 분석 시스템으로 테스트한 특징추출 결과이다 (그림 21)은 얼굴 생성 시스템으로부터 생성되어진 결과물이다.



(그림 19) 특징 분석



(그림 20) 특징추출 결과

실험 후 분석한 성공률은 다음과 같다.

<표 2>는 얼굴영역 추출의 성공률이며, <표 3>은 각 이목구비 추출의 성공률을 도표화한 것이다. <표 3>에서 보는 바와 같이 눈의 위치 판별의 성공률은 비교적 높은 편(96%)이었으나 눈썹의 성공률은 그다지 높지 않은 것으로 나타났다. 그 이유는 눈썹은 머리카락에 가려지거나 열린 눈썹을 가진 사람이 많고 또 조명에 따라서도 영향을 받기 때문이다. 이는 모양 특징정보 추출 시에도 영향을 미치는 것으로 나타났다<표 4>. 코의 경우는 이목구비 중에서 가장 추출률이 낮은 것으로 나타났는데 여러모로 코의 특징추출이 어려움을 보여주고 있다. 모양정보 추출에 있어서도 57%의 낮은 성공률을 보였다. 이는 코가 특히 조명변화에 민감하다는 것을 입증한다. 코의 위치는 눈과 입의 위치로 추정할 수 있으므로 <표 3>에서는 빠져있다. 모양 특징정보 추출에 있어서 눈, 입의 경우는 윤곽선이 뚜렷하기 때문에 비교적 높은 성공률(각각 88%, 95%)을 보인 반면 눈썹, 코, 턱의 경우는 모양 추출에 있어서 낮은 성공률을 나타내었다. 단, 턱의 성공률이 70%를 넘어섰다는 점과 캐리커처 생성에 있어서 핵심인 눈의 성공률이 비교적 높은 88%를 기록했다는 점에서 고무적이라 할 수 있겠다<표 4>.

<표 2> 얼굴영역 추출 성공률

총 사진의 수	성공	실패
194	183(94.3%)	11(6.7%)

<표 3> 이목구비 추출 성공률(Localization)

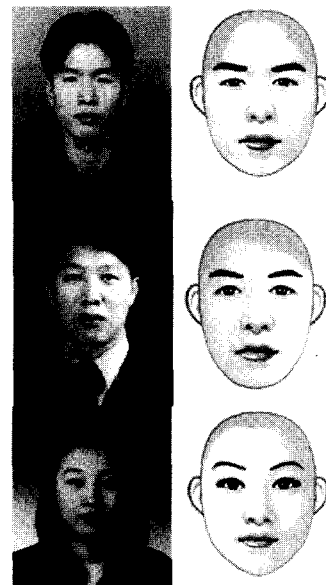
이목구비	전체 사진개수	양쪽 모두 성공	한쪽만 성공	실패
눈	183	161(88%)	15(8%)	7(4%)
눈썹	176	79(45%)	41(23%)	56(32%)
입	176	143(81%)		33(19%)

<표 4> 모양 특징정보 추출 성공률

이목구비	전체 개수	VERY GOOD	GOOD	BAD
눈	176	92(52%)	63(36%)	21(12%)
눈썹	120	39(22%)	79(45%)	58(33%)
코	176	28(24%)	39(33%)	51(43%)
입	143	77(54%)	59(41%)	7(5%)
턱	176	56(32%)	72(41%)	48(27%)

10. 결론 및 향후 연구

본 논문에서는 얼굴 사진을 이용하여 자신과 닮은 캐리커처를 자동으로 생성해 주는 시스템을 구현하였다. 이 시스템은 얼굴 사진으로부터 얼굴 영역만을 추출해 내는 모듈과 이목구비의 부분 영역을 추출하는 모듈과 각 이목구비별 특징정보를 추출하는 모듈 그리고 얻어진 특징정보를 이용해서 얼굴 모양(턱선)을 벡터화하고 사진으로부터 음영정보를 추출하여 음영판을 생성하는 모듈, 최종적으로 얼굴형과 특징정보를 이용하여 DB로부터 검색되어진 이목구비를 특징정보에 맞게끔 적당한 위치에 배치하여 캐리커처의 얼굴을 생성하는 모듈을 포함하고 있다.



(그림 21) 얼굴생성 결과

논문에서 제시하는 시스템은 벡터기반이기에 자유로운 변형이 가능하다는 점과 윤곽선만을 이용하던 지금까지의 논문들과 달리 컬러 및 음영을 결과물에 적용하고, 또한 디자이너 또는 그림 작가의 느낌을 결과물에 그대로 지닐 수 있도록 한다는 점에서 타 논문과의 차별성을 갖는다.

이 시스템을 통해 특정 인물의 사진을 이용하여 캐릭터를 자동으로 생성하는 테스트를 수행하였다. 이에 대한 실험결과 및 성능평가 결과는 논문에 제시하였다.

향후 계획은 기울어진 얼굴과 측면 얼굴 그리고 흑인의 얼굴로도 캐리커처를 생성할 수 있도록 시스템을 개발해 나가는 것이다. 또한, 캐리커처에 과장(exaggeration) 개념을 도입하여 실제 캐리커처의 효과를 낼 수 있도록 할 계획이며, 사람의 표정도 흉내 낼 수 있도록 할 계획이다.

참 고 문 헌

[1] S. Brennan, "Caricature Generator," Master's thesis, Cambridge, MIT, 1982.

[2] Junji Nishino, Tomonori Kameyama, Haruhiko Shirai, Tomohiro Odaka and Hisakazu Ogura, "Linguistic Knowledge Acquisition System on Facial Caricature Drawing System," IEEE International Fuzzy Systems Conference Proceedings, pp.22-25, August, 1999.

[3] Jun-ichiro Hayashi, Kazuhito Murakami, Hiroyasu Koshimizu, "Facial Caricaturing System PICASSO with Emotional Motion Deformation," Second International Conference on Knowledge-Based Intelligent Electronic System, pp.21-23, April, 1998.

[4] Francesco, G. B., De Natale, Daniele, D. Giusto and Fabrizio Maccioni, "A Symmetry-Based Approach To Facial Feature's Extraction," IEEE, 1998.

[5] G. HANAOKA, M. KANEKO, H. HARASHIMA., "Facial caricature by computer based on the style of individual human caricaturist," IEICE trans, Vol.J80-D-II, No.8, pp. 2110-2118, 1997.

[6] S. Iwashita and T. Onisawa, "A study on facial caricature drawing by fuzzy theory," Proceedings of FUZZ-IEEE '97, pp.933-938, 1997.

[7] J. Nishino, M. Shimasaki, H. Shirai, T. Odaka and H. Ogura, "facial caricature drawing system based on a fuzzy linguistic expression," Proceedings of 13rd. Fuzzy System Symposium, pp.465-468, 1997.

[8] Tominaga, M., Fukuoka, S., Murakami, K. and Koshimizu, H., "Facial Caricaturing with Motion Caricaturing in PICASSO System," IEEE/ASME International Conference on Advanced Intelligent Mechatronics AIM'97, p.30, 1997.

[9] Jianming Hu, Hong Yan and Mustafa Sakalli, "Facial Feature Extraction from Head-Shoulder Images," The Fifth International Conference on Control, Automation, Robotics and Vision(ICARCV '98) Singapore, pp.9-11, December, 1998.

[10] Rein-Lien Hsu, A. K. Jain, M. Abdel-Mottaleb, "Face

detection in color images," IEEE Trans on Pattern Analysis and Machine Intelligence, Vol.24, No.5, pp.696-706, 2002.

[11] Ming-Hsuan Yang, D. J. Kriegman, N. Ahuja, "Detecting faces in images : a survey," IEEE Trans on Pattern Analysis and Machine Intelligence, Vol.24, No.1, pp.34-58, 2002.

[12] Hong Chen, Ying-Qing Xu, Heung-Yeung Shum, Song-Chun Zhu, and Nan-Ning Zheng, "Example-based Facial Sketch Generation with Non-parametric Sampling," Proc. International Conference on Computer Vision, 2001.

[13] R. Brunelli and T. Poggio, "Caricatural Effects in Automated Face Perception," Biological Cybernetics, pp.235-241, 1993.

[14] 조용진, 얼굴 한국인의 낮, 사계절 출판사, 1999.

[15] 이육경, "얼굴 특징정보를 이용한 캐리커처 자동 생성 시스템", 숭실대학교 석사학위 논문, 2002.

[16] 최내원, 오해석, "실시간 칼라영상에서 객체추출 및 추적", 정보처리학회논문지B, 제10-B권 제1호, pp.81-86, 2003.

[17] 박연출, 오해석, "벡터기반의 캐리커처 자동생성에 관한 연구", 정보처리학회논문지B, 제10-B권 제6호, 2003.

[18] E. Hjelmas, B. K. Lee, Face Detection : A Survey, "Computer Vision and Image Understanding," Vol.83, pp.236-274, 2001.

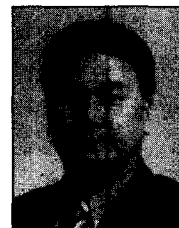
[19] M. H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting Faces in Images : A Survey," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.24, No.1, pp.34-58, Jan., 2002.

[20] N. Tsapatsoulis and S. Kollias, "Face Detection in Color Images and Video Sequences," Melecon 2000, Limassol Cyprus, 2000.



박 연 출

e-mail : parkpd@edunet.net
 1997년 상지대학교 응용통계학과(이학사)
 1999년 숭실대학교 컴퓨터학과(공학석사)
 2001년 숭실대학교 컴퓨터학과(박사수료)
 현재 숭실대학교 컴퓨터학과 박사과정
 관심분야 : 컴퓨터비전, 얼굴인식, Facial Caricaturing



오 해 석

e-mail : oh@kyungwon.ac.kr
 서울대학교 응용수학과(학사)
 서울대학교 계산통계학과(석사, 박사)
 일본 동경대학 객원 교수
 미국 스탠퍼드대학교 객원 교수
 한국 정보처리학회 회장 역임
 1982년~2003년 숭실대학교 컴퓨터학과 교수/부총장 역임
 2003년~현재 경원대학교 소프트웨어대학 교수/부총장
 관심분야 : Multimedia, Database, 지식경영