

규칙 기반 캐리커처 자동 생성 기법*

이은정^o, 권지용, 이인권

연세대학교 컴퓨터과학과

(angui^o, mage, iklee)^o@cs.yonsei.ac.kr

Automatic Generation of Rule-based Caricature Image

Eun-Jung Lee^o, Ji-yong Kwon, In-Kwon Lee

Dept. of Computer Science, Yonsei University

요약

본 논문은 주어진 얼굴 사진에 대하여 자동으로 얼굴의 특징을 강조한 캐리커처 생성 기법을 제안한다. AAM(Active Appearance Model)을 사용하여 트레이닝 이미지의 특징점과 텍스처 정보를 유지하고 이것을 이용하여 평균 얼굴의 정보와 함께 주어진 얼굴에 대한 특징점을 찾아낸다. 캐리커처 아티스트들의 제안을 바탕으로 특징적인 부분을 과장하기 위한 룰을 정의하고 이를 입력 얼굴의 특징점에 적용하여 과장된 특징점을 얻는다. 마지막으로 주어진 사진에 대하여 좀 더 만화적인 효과를 내기 위해 얼굴 이미지에 카투닝을 적용한 다음 과장된 특징점으로 와평한다. 이러한 방법으로 사용자의 조작을 최소화 하는 캐리커처 생성을 할 수 있다.

Abstract

We present the technique that automatically generates caricatures from input face images. We get the mean-shape of training images and extract input image's feature point using AAM(Active Appearance Model). From literature of caricature artists, we define exaggeration rules and apply our rules to input feature points, then we can get exaggerated feature points. To change our results into cartoon-like images, we apply some cartoon-stylizing method to input image and combine it with facial sketch. The input image is warped to exaggerated feature point for final results. Our method can automatically generate a caricature image while it minimizes user interaction.

키워드(국문): 캐리커처, 만화화, 동적 외양 모델

Keywords(English): caricature, cartooning, AAM

1. 서론

최근 컴퓨터 그래픽스 기술의 발달과 사용자 맞춤형 서비스에 대한 관심이 증가하면서 캐리커처 생성 기법에 대한 연구가 활발하게 이루어 지고 있다. 그러나 캐리커처를 그리는 것은 여전히 아티스트에 의존적인 경향이 있다. 일부 서비스업체에서는 온라인으로 캐리커처를 제공해주기도 하지만 그 역시 아티스트가 직접 그리기 때문에 시간이 걸린다. 그래서 대부분의 인터넷서비스 업체에서는 아바타를 자신의 얼굴을 대신하는 경우가 많다. 그러나 아바타는 이미 정해져 있는 객체들을 조합하기 때문에 실제 얼굴의 표현이라고 할 수 없다.

이에 우리는 스케치에 익숙하지 않은 사람도 쉽게 캐리커처를 생성할 수 있는 방법을 제안한다. 우리가 제안하는 방법은 여러 얼굴데이터의 특징점을 바탕으로 평균얼굴을 구하고 새로운 얼굴과 평균얼굴과의 비교를 통하여 자동으로

과장의 정도를 계산하며, 이를 바탕으로 특징이 강조된 캐리커처를 아티스트의 도움 없이 생성할 수 있다.

본 논문의 순서는 다음과 같다. 2절에서는 캐리커처 생성에 대한 기존의 연구를 살펴본다. 3절에서는 캐리커처를 생성하기 위한 전체적 흐름을 설명하고 알고리즘의 각 부분을 자세히 설명하겠다. 4절에서는 제안한 기법을 통해 생성해 낸 결과를 제시하고 5절에서 결론을 맺겠다.

2. 관련연구

캐리커처에 관한 연구는 크게 두가지로 나눌수 있다. 사용자가 직접 과정에 참여하는 연구가 있었고 자동으로 과장의 정도를 추출하고 반영하는 연구가 있었다.

• 수동 과장

Akleman[9]은 라인 세그먼트를 컨트롤 하는 방식으로 캐리커처를 생성하였다. 사용자가 라인의 위치를 움직여서 과장을 적용하였다. Koshimizu[11]는 웹에서 동작

*본 연구는 한국전자통신연구원의 정보통신연구개발사업 위탁연구과제지원으로 이루어졌음.

하는 Web-PICASSO를 통해 템플릿을 기반으로 캐리커처를 생성하였다. 이는 사용자가 과장의 비율을 조절하여 적절한 결과를 만들어낸다. 수동으로 과장을 하면 만족도가 높은 캐리커처를 생성해 낼 수 있지만 일반적인 사람들과 다른 특징을 강조하는 캐리커처의 기본적인 목적에서 벗어날 수 있다.

• 자동 과장

Liang[7]은 자동으로 과장을 하기 위해서 아티스트의 성향을 반영하는 예제기반 방식으로 접근을 하였다. 아티스트가 직접 그린 캐리커처를 기반으로 생성하기 때문에 과장이 자연스럽게 되지만 예제를 만들기 위해 많은 캐리커처를 그려야 하고 예제로 사용된 캐리커처를 그린 아티스트의 성향에 의존하는 경향이 있다. Chiang[8]은 MPEG-4 표준에 맞게 얼굴의 특징을 정의하고 아티스트가 미리 그려놓은 캐리커처의 특징점으로 와핑을 적용해서 캐리커처를 생성하였다. 기준이 되는 캐리커처를 다양하게 해줌으로 여러 가지 스타일을 만들 수 있지만 이 방법 역시 아티스트에 의존하게 된다.

• 반자동 과장

Gooch[4]는 캐리커처를 그리기 위해 Redman[2]이 제안한 인비트워너를 사용하고 있다. 그는 인비트워너를 통해 얼굴의 비율을 계산하여 과장을 적용한다. Gooch의 시스템은 과장을 위해 자동으로 비율을 계산해 주기는 하지만 인비트워너를 구성하기 위해 사용자가 관여해야 하는 점에서 완전히 자동시스템은 아니다.

본 논문에서는 사용자의 상호작용을 최소로 하는 캐리커처 자동생성 기법을 제안한다. 우리가 제안하는 방법은 Redman이 제안한 인비트워너를 사용하지만 사용자의 관여 없이 AAM을 사용하여 자동으로 인비트워너를 생성한다. 그리고 캐리커처 아티스트들의 노하우를 몇 가지 규칙으로 정하여 아티스트들의 스타일을 반영한다.

3. 캐리커처 생성 기법

3.1 전체적인 알고리즘

그림 1은 우리가 제시하는 방법의 전체적인 흐름을 보여주고 있다. 먼저 AAM을 사용하여 입력 이미지에 대한 특징점을 찾는다. 입력된 얼굴의 특징점과 AAM을 구축하면서 계산한 평균얼굴의 특징점을 비교하여 얼굴을 과장하게 된다. 과장은 크게 두 가지 단계로 진행된다. 우선, 인비트워너를 사용하여 얼굴의 전체의 비율을 과장한다. 그 다음 얼굴의 컴퍼넌트에 해당하는 특징점에 대하여 각각의 컴퍼넌트의 특징을 과장한다. 최종적으로 이미지의 만화적인 스타일을 위해 스케치와 카투닝을 한 뒤 이를 과장된 특징점으로 와핑하여 입력 이미지에 대한 캐리커처를 생성한다.

3.2 평균얼굴 및 특징점 검출

AAM(Active Appearance Model)[10]은 물체의 형태와 텍스처 정보를 학습시켜 이 정보를 바탕으로 새로운 이미지 내의 물체를 찾는 방법이다. 우리는 AAM을 구축하기 위해서 20장

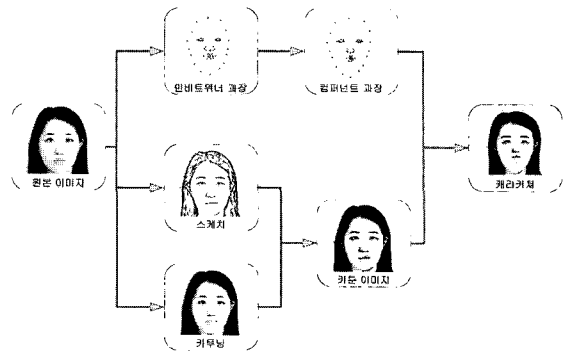


그림 1: 알고리즘의 전체적인 흐름도

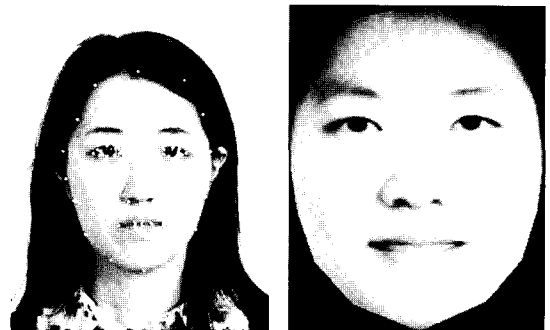


그림 2: 트레이닝 데이터와 평균얼굴

의 트레이닝 이미지를 사용하였고 각각의 이미지에 대해서 미리 정해둔 54개의 특징점을 입력하였다. 특징점의 입력에는 공개소프트웨어를 사용하였다. 우리는 입력된 특징점을 바탕으로 AAM을 구축하였고 트레이닝 이미지에 대한 평균얼굴을 생성하였다(그림 2). AAM을 통해서 우리는 새로운 인풋이미지에 대한 특징점을 검출할 수 있다(그림 2).

좀 더 명확한 특징점 검출을 위해서 우리는 다음과 같은 조건을 만족하는 이미지를 사용하였다.

- 얼굴은 앞을 바라보고 있어야 한다. (모자나 안경 착용 금지)
- 모든 트레이닝 이미지는 같은 조명을 받아야 한다.

AAM을 이용해 특징점을 검출 할 경우, 특징점 검출 알고리즘은 이미지의 텍스처 정보를 사용하게 됨으로 일정한 조명을 유지 하였을 경우 더 정확한 특징점을 찾을 수 있다.

3.3 과장

우리는 입력된 이미지의 특징점과 평균얼굴의 특징점을 과장하는 방법을 두 가지 단계로 나누었다. 먼저 얼굴의 전체적인 비율을 과장하기 위해 인비트워너를 사용하였고, 얼굴



그림 3: Redman의 인비트위너

내부의 컴퍼넌트의 모양을 과장하기 위해 몇 가지 규칙을 만들어 적용하였다.

특징점에 과장을 적용하기 전에 입력된 얼굴의 회전정보를 계산해주어야 할 필요가 있다. 입력된 얼굴이 기울어져서 들어왔을 경우 평균얼굴과의 비교가 무의미해질 수 있기 때문이다. 회전정보를 구하는 방법은 다음과 같다.

$$\hat{x}_i = x_i - \mu_x, \quad \hat{x}_{mean,i} = x_{mean,i} - \mu_{x_{mean}}$$

$$\hat{x}'_i = s \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \hat{x}_i = \begin{bmatrix} a & -b \\ b & a \end{bmatrix} \hat{x}_i = \mathbf{T}\hat{x}_i$$

x_i 는 특징점을 나타내고 μ_x 는 특징점들의 평균점을 나타낸다. $x_{mean,i}, \mu_{x_{mean}}$ 는 평균 얼굴의 특징점과 특징점의 평균을 나타낸다. 입력 이미지의 특징점들(\hat{x}_i)이 최대한 평균 얼굴의 특징점들($\hat{x}_{mean,i}$)과 같아질 수 있는 회전행렬을 구한다. 여기서 최소화해야 하는 식은 다음과 같은 최소자승식으로 표현할 수 있다.

$$\text{minimize } O(\mathbf{T}) = \sum_i \|\mathbf{T}\hat{x}_i - \hat{x}_{mean,i}\|^2$$

이 최소자승식은 \mathbf{T} 를 이루는 두 문자 a 와 b 에 대한 2차식이므로, 각각에 대하여 편미분하면 다음과 같은 일반해를 얻을 수 있다.

$$a = \frac{\sum_i \hat{x}_i \cdot \hat{x}_{mean,i}}{\sum_i \|\hat{x}_i\|^2}, \quad b = \frac{\sum_i \det([\hat{x}_i \quad \hat{x}_{mean,i}])}{\sum_i \|\hat{x}_i\|^2}$$

$$s = \sqrt{a^2 + b^2}, \quad \theta = \cos^{-1}\left(\frac{a}{s}\right)$$

위의 식으로부터 계산한 θ 를 입력된 얼굴의 모든 특징점에 대하여 적용한다. 과장이 끝난후에는 다시 $-\theta$ 만큼 회전하여 원래의 각도를 유지하도록 한다.

3.3.1 인비트위너(In-betweener) 과장

Redman은 그의 저서 *How To Draw Caricature*에서 캐리커처의 과장을 위한 요소를 찾기 위한 방법으로 상대성원리를 제시하였다[2]. 얼굴의 눈, 코, 입의 크기, 위치등의 특징을 다른 사람들과 비교를 통해 얻는 것이다. 인비트위너는 얼굴에 대해서 정의된 비례모델이라고 할 수 있다(그림3). 우리는 특징점을 사용하여 인비트위너를 구축하였고 이것을 통해 얼굴의 비례정보를 추출하였다. 그리고 평균얼굴의 인비트위너와의 비교를 통해 과장을 적용하였다.

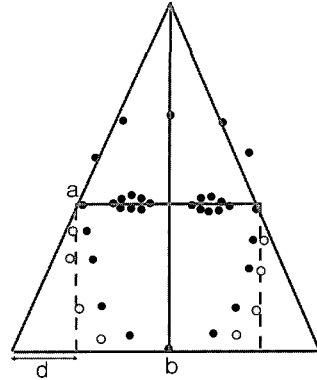


그림 4: 얼굴형태의 과장: 검은색점은 입력 얼굴의 원본 특징점, 회색점은 과장된 특징점을 나타낸다.

3.3.2 컴퍼넌트 과장

앞에서 설명한 인비트위너 과장을 통해서서는 각 컴퍼넌트의 형태에 관한 과장을 할 수는 없다. 예를 들어 가름하거나 둥그란 얼굴형태, 눈이 올라가거나 처진 형태는 얼굴의 비율을 통해서 강조할 수 없다. 우리는 컴퍼넌트의 형태를 과장시키기 위해서 규칙을 정의하였다. 각각의 컴퍼넌트 마다 정의한 규칙은 다르지만 평균얼굴과의 비교를 통해 과장을 하는 점에서 인비트위너 과장과 비슷한 방법이라고 할 수 있다.

• 얼굴형태

머리 부분을 구성하는 데에 있어, 두 개의 사다리꼴을 사용하는 것이 유용하다. 첫번째 사다리꼴은 머리끝에서 눈까지를, 두번째는 눈에서 턱까지의 부분을 구성하는 것이 일반적이다[3]. 우리는 눈을 기준으로 머리를 아래위로 나누었고 각각의 부분을 따로 과장하였다. 그림4는 얼굴의 아래부분을 사다리꼴로 표현한 것이다. 왼쪽 턱의 경우, 이 영역에 속하는 점들의 평균점으로부터 a, b 를 연결한 선과의 거리와 평균얼굴에서 같은 방법으로 구한 거리간의 비교를 통해서 가름하거나 둥근 정도를 알아낸다. 그 다음, 미리 정해진 과장 정도를 나타내는 상수 d 와 비례식을 통해서 원본 특징점(검은색 점)을 과장된 특징점(회색 점)의 위치로 변환하였다. 본 논문에서는 d 를 20으로 하였다.

• 눈

눈의 형태를 강조하기 위해 두 가지 규칙을 정의하였다.

1) 눈의 크기 과장: 눈의 가로, 세로 비율을 바탕으로 둥그란 모양 혹은 길쭉한 모양인지 판단한 뒤, 눈 전체에 크기조절을 적용시킨다. 왼쪽, 오른쪽 눈에 해당하는 특징점들을 바탕으로 각각 크기조절을 해준다.

$$\hat{x}_i = (x_i - x_{mean}) * ratio * s + x_{mean}, x_i \in E$$

이때 E 는 한쪽 눈에 해당하는 특징점들의 집합을, x_{mean} 은 집합 E 에 대한 평균점이다. $ratio$ 는 입력 얼굴

의 눈 크기와 평균얼굴의 눈 크기의 비율을 나타낸다. s 는 크기조정 상수로써 본 논문에서는 0.7을 사용하였다.

2) 눈의 각도 과장 : 눈 꼬리 부분의 특징점의 위치를 바탕으로 눈 꼬리의 처짐과 올라간 정도를 파악한 뒤, 눈 전체에 대해 회전을 적용시킨다.

$$\hat{x}_i = (x_i - x_{mean}) \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} + x_{mean}, x_i \in E$$

θ 는 입력 얼굴의 눈의 각도와 평균얼굴의 눈의 각도의 차이를 나타낸다.

• 코

코의 형태를 강조하기 위해 두 가지 규칙을 정의하였다.

1) 코의 크기 과장 : 코의 좌우 너비를 바탕으로 전체 얼굴에 대한 코의 상대적인 크기를 측정한 뒤 과장한다. 눈 크기의 과장과 같은 방법으로 수행한다.

2) 코의 형태 과장 : 코끝의 특징점을 기준으로 매부리코, 들창코의 여부를 판단하여 과장한다.

$$x = (x - x_{mean}) * ratio * s + x_{mean}$$

x 는 코끝을 나타내는 점, x_{mean} 은 코를 나타내는 특징점의 평균이다. $ratio$ 는 입력얼굴의 코끝점과 평균얼굴의 코끝점을 비교한 값이고 s 는 크기조정 상수로써 본 논문에서는 0.9를 사용하였다.

• 입

입의 형태를 강조하기 위해 두 가지 규칙을 정의하였다

1) 입의 크기 과장 : 입의 좌우 너비를 바탕으로 얼굴 전체에서의 비율을 측정한 뒤 과장한다. 눈 크기의 과장과 같은 방법으로 수행한다.

2) 입 꼬리의 과장 : 입 꼬리가 올라가거나 내려간 정도를 측정한뒤 좌우 입술의 특징점에 shear변환을 적용한다.

$$\hat{x}_i = (x_i - x_{mean}) \begin{bmatrix} 1 & 0 \\ ratio * s & 1 \end{bmatrix} + x_{mean}, x_i \in M$$

이때 M 는 입에 해당하는 특징점들의 집합을, x_{mean} 은 집합 M 에 대한 평균점이다. $ratio$ 는 입력얼굴에서 x_{mean} 과 입 꼬리의 점을 이은 선분과 평균얼굴에서 같은 방법으로 계산한 선분이 이루는 각도를 나타낸다. s 는 크기조정 인자로써 본 논문에서는 -0.005를 사용하였다.

3.4 스케치 및 카투닝

• 스케치

De Beeck은 입체감 있고 실감나는 캐리커처를 위해서 얼굴의 주름을 나타내는 것이 좋다고 하였다. 그리고 머리를 캐리커처할 때 두발 전체의 모습을 그리는 것도 중요하다[3].

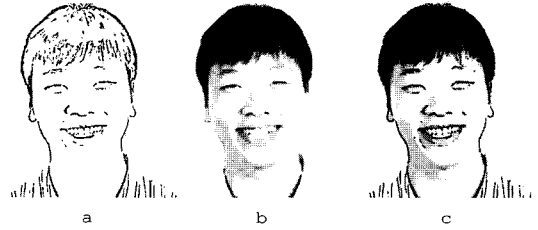


그림 5: 스케치와 카투닝 : a는 DOG를 이용한 스케치이고 b는 Bilateral 필터 적용 후 양자화한 이미지이고 c는 b에 a를 덧입힌 결과이다.

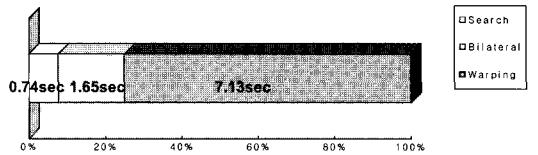


그림 6: 단계별 수행 시간

우리는 얼굴의 외곽선, 주름, 머리카락 등을 그리기 위해서 에지영역을 검출하여 스케치로 사용하였다. 보편적으로 에지는 높은 대비를 나타내는 곳에서 검출되므로 얼굴에서의 주름이나 머리카락, 얼굴 전체의 모습을 표현할 수 있다. 에지를 검출하기 위해 가우시안의 차이(DOG)를 이용하였다[5]. 두 가우시안의 커널 사이드를 조절하면 에지의 두께를 조절할 수 있어 스케치의 스타일에 변형을 줄 수 있다.

• 카투닝

캐리커처의 만화적 스타일을 위해 Bilateral 필터[6]를 적용하였다. 카툰화를 위해 여러 번의 Bilateral 필터를 수행하여 색상을 보다 단순화 시키고 최종적으로 양자화를 통해 이미지를 좀 더 만화적으로 표현하였다.

우리는 카투닝된 이미지에 스케치를 덧입혀서 입력 이미지를 카툰화하였다(그림5). 최종적으로 과장이 적용된 특징점으로 와핑을 적용하여 캐리커처를 얻을 수 있다. 우리는 주어진 특징점을 바탕으로 와핑하기 위해 특징선 기반 와핑 기법[13]을 사용하였다.

4. 결과

실험을 위해 우리는 상용소프트웨어를 사용하여 각각의 트레이닝 이미지에 54개의 특징점 입력하여 AAM을 구성하였다. AAM의 구성을 위해 공개 라이브러리인 AAM-API를 사용하였다[1]. 캐리커처를 생성하기 위해 평균적으로 10초 정도의 시간이 소요되었다. 단계별로 수행하는데 소요되는 시간은 특징점을 찾는 데 0.74초, 스케치와 Bilateral 필터가

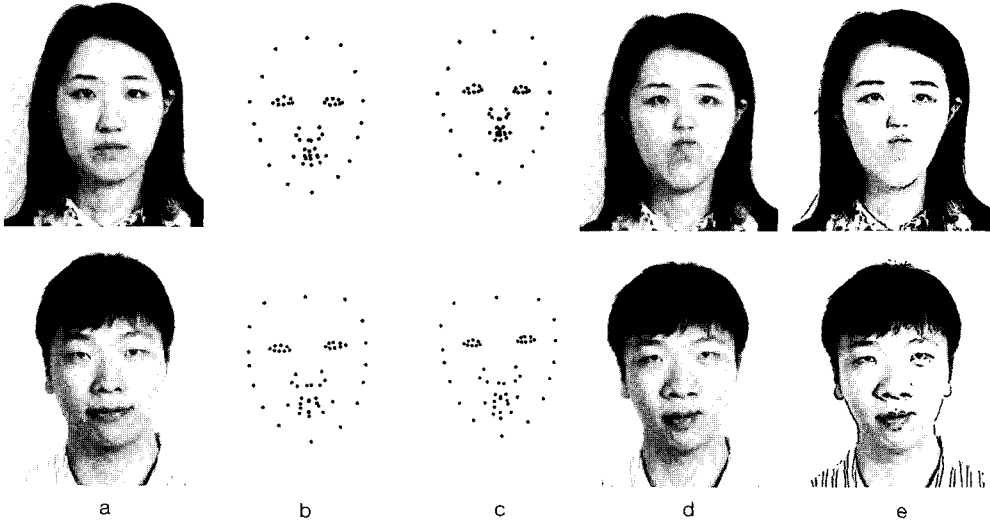


그림 7: 과장과 카투닝을 적용한 결과: a는 원본 이미지이고 b는 원본 이미지에서 추출한 특징점이다. c는 과장을 거친후의 특징점이고 d는 과장된 이미지, e는 과장된 카툰이미지다.

1.65초에 수행되었고 마지막으로 와핑에 7.13초가 소요되었다(그림6).

그림7는 우리의 알고리즘으로 생성된 캐리커처이다. (a)는 원본 이미지이고 이미지에서 얼굴의 특징점을 추출한 것이 (b)이다. (c)는 우리의 알고리즘을 통해 과장된 특징점이고 (d)는 이미지를 와핑시킨 것이다. 마지막으로 (e)는 카투닝을 시킨 결과이다. 실험을 통해서 우리가 제안한 알고리즘은 입력된 얼굴에 대하여 평균얼굴과 차이가 많이 나는 컴퍼넌트를 더욱 강조하는 것을 확인할 수 있었다. 또한 Bilateral 필터와 DOG 필터를 통해 얻어낸 카툰형식의 이미지를 와핑하여 결과를 생성함으로써 카툰의 느낌이 나는 캐리커처 이미지를 얻을 수 있었다.

5. 결론

우리는 캐리커처를 자동으로 생성하기 위한 방법을 제시하였다. 제시한 방법을 통하여 우리는 평균얼굴과 입력된 얼굴을 비교하여 기존에 제시된 방법과 같은 비례적인 과장뿐만 아니라 특징이 되는 컴퍼넌트가 강조된 캐리커처를 얻을 수 있었다. 또한 특징점의 추출부터 비교, 과장하는 단계까지 모두 자동으로 이루어지므로 사용자의 관여 없이 결과를 생성할 수 있었다.

그러나 우리가 제시한 방법에는 몇 가지 제한점이 있다. 입력 이미지에서 특징점을 추출하기 위해 사용하는 AAM이 항상 정확한 특징점을 찾아주지는 않는다. 이러한 단점을 극복하기 위해 우리는 조명을 일정하게 유지해주는 방법을 선택하였지만 언제나 올바른 결과를 얻는다는 보장은 없다. 한편 텍스처의 영향을 많이 받지 않는 ASM(Active Shape Model)을 고려해볼 수 있으나, ASM의 경우 AAM에 비해 더

많은 특징점을 가지고 있어야 하므로, 데이터베이스의 구축이나 와핑 속도에 더 안 좋은 영향을 줄 수 있을 것이다. 현재 이와 관련된 많은 연구가 진행되고 있기 때문에 좀 더 개선될 수 있을 것이다. 또한 컴퍼넌트의 과장을 위해 사용한 간단한 비례식을 좀 더 다양한 방식으로 대체하면 보다 질 좋은 결과물을 생성할 수 있을 것이다. 마지막으로 보다 빠른 와핑방법을 사용하면 실시간에 결과를 얻어낼 수 있을 것이다.

참고 문헌

- [1] AAM Library , <http://www2.imm.dtu.dk/~aam>.
- [2] Lenn Redman , *How To Draw Caricature Contemporary Books*, 1984.
- [3] Jan Op De Beeck , *L'Art de la caricature B and B Press*, 2001.
- [4] Bruce Gooch, Erik Erinhard and Amy Gooch , *Human Facial Illustrations:Creation and Psychophysical Evaluation ACM Transactions on Graphics*, 2004.
- [5] Holger Winnemoller, SvenC.Olsen and Bruce Gooch , *Real-Time Video Abstraction SIGGRAPH '06 Proceeding*, 2006.
- [6] Tomasi.C, Manduchi.R , *Bilateral Filtering for Gray and Color Images ICCV '98 Proceeding*, 1998.

- [7] L.Liang,H.Chen,Y.Q.Xu and H.Y.Shum , Example-Based Caricature Generation with Exaggeration *10th Pacific Conference, on Computer Graphics and Applications Proceeding* , 2002.
- [8] P.Y.Chiang, W.H.Liao and T.Y.Li , Automatic Caricature Generated by Analyzing Facial Features *ACCV '04 Proceeding*, 2004.
- [9] E.Akleman , Making Caricature with Morphing *Siggraph '97 Proceeding*, 1997.
- [10] T.F.Cootes, C.J.Taylor , Statistical Models of Appearance for Computer Version *Technical Report, University of Manchester* , 2000.
- [11] H.Koshimizu, M.Tominaga, T.Fujiwara and K.Murakami , On Kansei Facial Processing for Computerized Facial Caricaturing System Picasso *Proceeding of IEEE International Conference on Systems,Man,and Cybernetices* , 1999.
- [12] S.Brennan , Caricature Generator *Master's thesis, Cambridge, MIT*, 1982.
- [13] T.Beier, S.Neely , Feature-Based Image Metamorphosis *Siggraph '92 Proceeding*, 1992.