

# 간단한 사용자 인터페이스에 의한 벡터 그래픽 캐릭터의 자동 표정 생성 시스템

박태희<sup>†</sup>, 김재호<sup>\*\*</sup>

## 요 약

본 논문에서는 가우시안 프로세스 모델을 이용한 벡터 그래픽 캐릭터의 자동 표정 생성 시스템을 제안한다. 제안한 방법은 Russell의 내적 정서 상태의 차원 모형을 근거로 재정의된 캐릭터의 26가지 표정 데이터로부터 주요 특징 벡터를 추출한다. 그리고 추출된 고차원의 특징 벡터에 대해 SGPLVM이라는 가우시안 프로세스 모델을 이용하여 저차원 특징 벡터를 찾고, 확률분포함수(PDF)를 학습한다. 확률분포함수의 모든 파라메타는 학습된 표정 데이터의 우도를 최대화함으로써 추정할 수 있으며, 이는 2차원 공간에서 사용자가 원하는 얼굴 표정을 실시간으로 선택하기 위해 사용된다. 시뮬레이션 결과 본 논문에서 제안한 표정 생성 프로그램은 얼굴 표정의 작은 데이터셋에도 잘 동작하며, 사용자는 표정과 정서간의 관련성에 관한 사전 지식이 없이도 연속되는 다양한 캐릭터의 표정을 생성할 수 있음을 확인할 수 있었다.

## Automatic facial expression generation system of vector graphic character by simple user interface

Tae Hee Park<sup>†</sup>, Jae Ho Kim<sup>\*\*</sup>

## ABSTRACT

This paper proposes an automatic facial expression generation system of vector graphic character using gaussian process model. Proposed method extracts the main feature vectors from twenty-six facial data of character redefined based on Russell's internal emotion state. Also by using new gaussian process model, SGPLVM, we find low-dimensional feature data from extracted high-dimensional feature vectors, and learn probability distribution function(PDF). All parameters of PDF are estimated by maximization the likelihood of learned expression data, and these are used to select wanted facial expressions on two-dimensional space in real time. As a result of simulation, we confirm that proposed facial expression generation tool is working in the small facial expression datasets and can generate various facial expressions without prior knowledge about relation between facial expression and emotion.

**Key words:** vector graphic character(벡터 그래픽 캐릭터), internal emotion state(내적 정서 상태), SGPLVM(가우시안프로세스 잠재 벡터모델), PDF(확률분포함수), likelihood(우도)

## 1. 서 론

표정은 자신의 감정 뿐만 아니라 상대방의 정서나

감정을 가장 빠르게 표현할 수 있는 매우 중요한 수단이다. 따라서 얼굴 애니메이션은 컴퓨터 그래픽스 분야에서 가장 활발하게 연구되고 있으며, 지금까지

※ 교신저자(Corresponding Author) : 박태희, 주소 : 부산시 남구 신선로 179번지(608-711), 전화 : 051)629-1538, FAX : 051)629-1529, E-mail : thpark77@dreamwiz.com

접수일 : 2009년 2월 16일, 완료일 : 2009년 6월 1일

<sup>†</sup> 정회원, 동명대학교 메카트로닉스공학과 전임강사

<sup>\*\*</sup> 정회원, 부산대학교 공과대학 전자공학과 정교수 (E-mail : jhkim@pusan.ac.kr)

※ 본 논문은 부산대학교 자유과제 학술연구비(2년)에 의하여 연구되었음, 2단계 BK21 사업의 지원으로 수행되었음

얼굴 표정을 생성하기 위한 많은 방법들[1-4]이 연구되고 있다.

얼굴 표정에 관한 대표적인 연구로서 Plutchik[5]는 문화적 보편성에 기반하여 8가지 순수 감정을 분노, 두려움, 혐오, 놀람, 기쁨, 슬픔, 수용, 기대감으로 나누고 그에 따른 표정을 분류하였으며, 순수 감정의 조합으로 복합 감정을 나타내는 감정 모델을 제안하였다. 또한 Ekman[6]은 얼굴을 통해 보편적으로 인지하거나 표현할 수 있는 6가지 기본 감정을 행복, 분노, 공포, 슬픔, 혐오, 놀람으로 분류하였다.

그러나 이러한 모델은 감정간의 상관관계를 표현하기 어렵고 감정의 연속된 변화나 미세한 변화를 표정에 반영하는데 제한이 있다. 이에 Russell[7]은 감정 차원을 “쾌/불쾌, 각성/비각성”의 두가지 차원으로 구조화하고, 연속적인 감정이나 감정의 강도에 따라 표정이 구현되도록 하였다. 또한 Ekman[8]은 FACS(Facial Action Coding System)를 제안함으로써 얼굴에 대한 감정과 표정을 근육의 움직임인 Action Unit의 조합으로 정의하고, 복잡하고 미세한 근육의 움직임만을 표기함으로써 효율적으로 표정의 구조를 정의하였다.

하드웨어와 3차원 소프트웨어가 발달함에 따라 Parkel[9]는 최초로 컴퓨터를 이용한 얼굴 모델의 3차원 표정 생성 및 형태 변화 시스템을 제안하였다. 이는 제안된 시스템에서 몇 개의 파라미터를 정의하고 이를 사용하여 얼굴의 필요한 요소를 변화시킴으로써 표정 및 형태를 변화시켰다. 또한 Waters[10]는 근육 벡터와 동적 파라미터를 제안함으로써 각 근육에 대해 근육의 영향이 미치는 범위, 최대 움직임 등을 정의하고 각 근육이 움직임에 따라 그와 연관된 피부의 점들이 움직이도록 하였다.

그러나 3차원 애니메이션은 입체감과 사실성이 우수한 반면 작업 시간과 비용, 인력이 많이 요구된다. 따라서 몇년 전부터 인터넷 환경에서는 플래시를 이용한 2차원 웹 애니메이션(Web Animation)이 활성화되기 시작하였으며[11], 최근에는 모바일 기기에서의 2차원 그래픽의 요구가 증가하고 있다. 또한 2차원 벡터 그래픽 기술 표준인 SVG(Scalable Vector Graphics)가 OpenVG로 표준화되면서[12] 데이터량은 작지만 확대시 화질 저하가 없고 회화적이며 질감 표현이 우수한 2차원 애니메이션에 대한 요구가 높아지고 있다. 한동안 3차원 애니메이션에

매진한 월트디즈니사가 2009년 개봉 예정인 신작 “공주와 개구리”를 2차원 애니메이션으로 제작한 것도 의미가 있다[13].

2차원 애니메이션에 관한 연구로서 고혜영[14]은 Russell의 표정 차원에서 제시된 9가지 실사 사진과 가장 잘 대응되는 과장된 애니메이션 캐릭터의 표정을 재정의하고, 정서들 간의 연관성을 바탕으로 연속되는 표정 변화를 자연스럽게 구현할 수 있는 표정 생성 시스템을 제안하였다. 또한 웹 및 모바일 서비스, 게임 등에서 실시간 표정 생성을 위해 주성분 분석(Principle Component Analysis)[15,16]에 기반한 표정 분석 및 생성에 관한 많은 방법들이 연구되고 있다. 그러나 이러한 방법들은 선형적인 표정 공간을 사용하므로 비선형적이고 멀티모달한 캐릭터의 표정을 생성하는 데는 제한이 있다.

따라서 본 논문은 사용자가 캐릭터의 비선형적인 표정을 자연스럽게 생성하는 데 있어 가장 간편한 사용자 인터페이스를 제공하는데 목적이 있다.

이를 위해 본 논문에서는 Russell의 내적 정서 상태의 차원 모형을 근거로 재정의된 캐릭터의 26가지 표정 데이터에 대해 SGPLVM(Scaled Gaussian Process Latent Variable Model) 확률 모델을 사용하여 비선형적인 캐릭터의 표정 변화를 분석한다. 그리고 분석된 데이터를 이용하여 캐릭터의 표정을 자연스럽게 구현할 수 있는 표정 자동 생성 시스템을 제안한다. 표정 학습에 사용된 벡터 그래픽 캐릭터들은 각 스타일에 따라 서로 다른 표정 공간을 가지게 된다. 그러나 다양한 캐릭터로의 확장과 사용자 인터페이스 구현의 단순화를 위해서는 통일된 표정 공간이 필요하다.

따라서 본 논문에서는 통일된 표정 공간을 학습하기 위해 변형된 최적화 수식을 제안하고, 캐릭터들의 각 스타일에 따른 몇 개의 파라미터들을 학습한다. 이로부터 캐릭터들의 표정 데이터는 고정된 2차원 공간 상에 분포되며, 각 스타일에 따라 학습된 파라미터들은 실시간으로 새로운 얼굴 표정을 생성하는데 사용된다. 또한 간단한 보간법을 통해 동일한 표정을 가지는 중간 캐릭터들을 생성하는 방법을 제안한다. 시뮬레이션 결과 사용자는 표정과 정서와의 관련성에 대한 사전 지식없이 중간 성격을 갖는 인물의 표정까지도 생성 가능함을 확인할 수 있었다.

본 논문의 구성은 다음과 같다. 2장에서는 벡터

그래픽의 원리와 SGPLVM에 대하여 설명하고, 3장에서는 본 논문에서 제안한 벡터 그래픽 캐릭터의 얼굴 표정 학습 및 생성에 대해 기술한다. 그리고 4장에서 실험 결과를 설명하고 5장에서는 결론 및 향후 과제에 대해 논의한다.

## 2. 관련 이론

### 2.1 벡터 그래픽의 원리

벡터 그래픽(vector graphics)은 픽셀을 기반으로 하는 비트맵 그래픽(bitmap graphics)과는 달리 기하학 정보를 수식으로 기술하여 최종 이미지를 재생하는 방식으로서 파일의 용량이 매우 작고 해상도에 따라서 깨지거나 계단 현상이 나타나지 않는다.

벡터 그래픽은 베지어 곡선(Bezier curve)[17]이라 불리는 매끄러운 곡선과 벡터 직선으로 구성되며, 제어점의 위치에 의해 곡선의 형태가 결정된다. 이때 제어점의 수는 베지어 다항식의 차수를 결정한다. 그림 1은 3차 베지어 곡선 세그먼트를 보인 것이다.

3차 베지어 곡선은 시작점  $P_0$ 와 끝점  $P_3$ , 곡선의 모양을 제어하는 두 개의 제어점  $P_1, P_2$ 로 정의된다. 시작점과 끝점은 선의 양쪽 끝에 위치하여 선의 길이를 결정하며, 나머지 두 개의 제어점을 이동시키으로써 곡선의 방향과 모양을 조절한다. 이때 방향선의 길이와 각도에 따라 곡선의 형태가 바뀌게 된다.

그림 2는 베지어 곡선의 조합으로 만들어진 벡터 그래픽 캐릭터의 얼굴을 보인 것이다[16].

그림 2의 벡터 그래픽 캐릭터는 총 321개의 제어점으로 이루어져 있으며, 눈동자 뿐만 아니라 입 속까지도 표현이 가능하도록 만들어졌다. 얼굴 표정을 표현하기 위해서는 이러한 제어점 좌표 뿐만 아니라 여러가지 제어 정보, 즉 각 오브젝트(눈, 코, 입 등)에 대한 레이어 정보와 크기 및 형태, 그리고 색상 정보 등으로 구성된다.

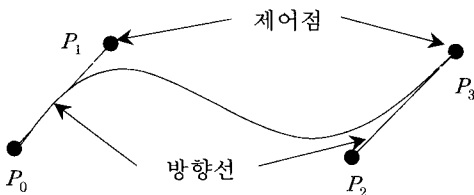


그림 1. 3차 베지어 곡선 세그먼트

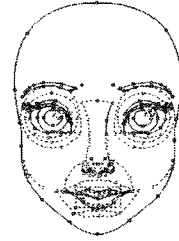


그림 2. 베지어 곡선의 조합에 의한 벡터 그래픽 캐릭터

이와같이 벡터 그래픽 캐릭터의 얼굴은 고차원의 데이터로 구성되므로 표정에 대한 유용한 모델을 학습하는데 있어서 큰 문제가 될 뿐만 아니라 다양한 캐릭터의 자연스러운 표정을 모든 제어점들을 조정하면서 만들어 나가는 것은 매우 어려운 일이다. 따라서 더욱 효율적인 표정 제어 및 생성을 위해 저차원 표정 데이터가 요구된다.

이를 위해 본 논문에서는 캐릭터의 표정으로부터 학습에 효과적인 특징 벡터를 추출한다. 학습에 사용된 특징 벡터  $\mathbf{ex}$ 는 각 표정의 제어 정보에 대응되는 각 오브젝트의 제어점 좌표와 크기 및 형태 정보로서 한 예로 본 실험에 사용되어진 얼굴 표정의 특징 벡터는  $D(=2880)$  차원의 정보로 구성된다.

### 2.2 SGPLVM

SGPLVM[18]은 가우시안 프로세스에 기반한 확률적인 차원 축소 모델로서 저차원 잠재공간(latent space)과 고차원 전공간(full space)간의 연속 매핑을 가능하게 하는 모델을 학습한다. 또한 전공간 벡터들에 대해 스케일링 인자(scaling factor)를 학습함으로써 비선형 데이터들의 서로 다른 스케일을 반영한다. 따라서 SGPLVM은 비선형적이고 멀티모달한 고차원 전공간의 표정 모델을 학습하기에 적합하다.

고차원 전공간의 표정(expression),  $\mathbf{ex}$ 와 저차원 잠재공간의 표정(reduced expression),  $\mathbf{rex}$ 를 연속적으로 매핑하기 위해 SGPLVM 알고리즘을 표정 관점에서 정리해 보면 다음과 같다.

만일  $i$ 번째 표정 벡터  $\mathbf{ex}_i$ 가 크기  $D \times 1$ 의 열벡터라 가정하면,  $N$ 개의 표정 벡터 집합  $\{\mathbf{ex}_i\}_{1 \leq i \leq N}$ 는 크기  $D \times N$ 의 행렬로 구성된다. 이때 표정 벡터들의 평균  $\mathbf{m}$ 은 다음과 같다.

$$\mathbf{m} = \frac{1}{N} \sum_{i=1}^N \mathbf{ex}_i \tag{1}$$

그리고 각 표정들 간의 상관 관계를 분석하기 위해 다음과 같이 AC 표정 벡터를 구하고, 이를 벡터  $\mathbf{aex}_i$ 로 표기한다.

$$\mathbf{aex}_i = [ex_{1,l} - \mathbf{m}_p, \dots, ex_{N,l} - \mathbf{m}_i]^T \quad (2)$$

여기서  $ex_{i,l}$ 는  $i$ 번째 표정  $\mathbf{ex}_i$ 에서의  $l$ 번째 벡터를 나타낸다.

또한  $\mathbf{K}$ 는  $N$ 개의 저차원 표정 벡터  $\{\mathbf{rex}_i\}$ 에 대한 커널 행렬(kernel matrix)이며, SGPLVM은 비선형 매핑을 위해 다음과 같이 RBF(Radial Basis Function) 커널을 사용한다.

$$\mathbf{K}_{i,j} = k(\mathbf{rex}_i, \mathbf{rex}_j) = \alpha e^{-\frac{\gamma}{2} \|\mathbf{rex}_i - \mathbf{rex}_j\|^2} + \delta_{\mathbf{rex}_i, \mathbf{rex}_j} \beta^{-1} \quad (3)$$

여기서  $\delta_{\mathbf{rex}_i, \mathbf{rex}_j}$ 는 Kronecker delta 함수로서 저차원 표정 벡터  $\mathbf{rex}_i$ 와  $\mathbf{rex}_j$ 가 같은 점이면 1, 그렇지 않으면 0으로 정의된다. 또한  $\alpha$ 는 두 표정의 상관도를 의미하며,  $\beta$ 는 예측시 존재하는 잡음의 양, 그리고  $\gamma$ 는 커널 함수의 퍼짐(spread) 정도를 나타낸다.

따라서 AC 표정 벡터의 집합  $\{\mathbf{aex}_i\}$ 에 대한 SGPLVM 모델의 우도 함수는 다음과 같이 정의된다.

$$p(\{\mathbf{aex}_i\} | \{\mathbf{rex}_i\}, \alpha, \beta, \gamma, \mathbf{w}) = \prod_{i=1}^D \frac{w_i^N}{\sqrt{(2\pi)^N |\mathbf{K}|}} \exp\left(-\frac{1}{2} w_i^2 \mathbf{aex}_i^T \mathbf{K}^{-1} \mathbf{aex}_i\right) \quad (4)$$

이는 각 표정 벡터의  $l$ 번째 차원에 스케일링 파라미터  $w_l$ 를 도입함으로써 가우시안 프로세스 모델의 우도 함수를 일반화한 것이다.

여기서  $\mathbf{w} = \{w_1, \dots, w_D\}$ 은  $D$ 차원의 표정 벡터에 대한 스케일링 파라미터 집합을 나타내며,  $\{\mathbf{rex}_i\}$ 는 저차원  $R$ 의 잠재공간 표정 벡터이다. 일반적으로  $R \ll D$ 이며, 모든  $\mathbf{rex}_i$ 는 저차원 공간 상에서 각 표정 벡터  $\mathbf{ex}_i$ 에 대응된다.

이러한 모델 파라미터  $\{\mathbf{rex}_i\}, \alpha, \beta, \gamma, \mathbf{w}$ 은 다음과 같이 식 (4)에 대한 부 로그-사후확률(negative log-posterior probability)을 각 파라미터에 관해 최소화함으로써 학습된다.

$$L_{GP} = -\ln p(\{\mathbf{rex}_i\}, \alpha, \beta, \gamma, \mathbf{w} | \{\mathbf{aex}_i\}) = \frac{D}{2} \ln |\mathbf{K}| + \frac{1}{2} \sum_{i=1}^D w_i^2 \mathbf{aex}_i^T \mathbf{K}^{-1} \mathbf{aex}_i + \frac{1}{2} \sum_{i=1}^N \|\mathbf{rex}_i\|^2 + \ln \frac{\alpha \beta \gamma}{\prod_{i=1}^D w_i^N} \quad (5)$$

이와같이 모델 파라미터  $\{\mathbf{rex}_i\}, \alpha, \beta, \gamma, \mathbf{w}$ 가 학습되고 나면 사용자(user)는 저차원 표정 벡터,  $\mathbf{rex}_{user}$ 를 사용자 인터페이스 상에서 손쉽게 선택할 수 있고 원하는(desired) 고차원 AC 표정 벡터,  $\mathbf{daex}$ 을 복구할 수 있다. 이때 사용자 선택의 저차원 표정 벡터  $\mathbf{rex}_{user}$ 와 원하는 고차원 AC 표정 벡터  $\mathbf{daex}$ 의 결합 확률밀도함수(joint probability density function)은 다음과 같이 주어진다.

$$p(\mathbf{rex}_{user}, \mathbf{daex} | \{\mathbf{rex}_i\}, \alpha, \beta, \gamma, \mathbf{w}, \{\mathbf{aex}_i\}) \propto \exp\left(-\frac{\mathbf{rex}_{user}^T \mathbf{rex}_{user}}{2}\right) \times \frac{\mathbf{w}^{N+1}}{\sqrt{(2\pi)^{(N+1)D} |\mathbf{K}|^D}} \times \exp\left(-\frac{1}{2} (\mathbf{K} \mathbf{j}^{-1} \mathbf{aex}_j \mathbf{w}^2 \mathbf{aex}_j)\right) \quad (6)$$

여기서  $\mathbf{aex}_j = [\mathbf{aex}_1, \dots, \mathbf{aex}_N, \mathbf{daex}]^T$ 는 학습에 사용된 AC 표정 벡터 집합  $\{\mathbf{aex}_i\}$ 와 원하는 고차원 AC 표정 벡터  $\mathbf{daex}$ 의 결합(joint) 벡터이며,  $\mathbf{K}_j$ 는 다음과 같이 이에 상응하는 커널 행렬을 의미한다.

$$\mathbf{K}_j = \begin{bmatrix} \mathbf{K} & \mathbf{k}(\mathbf{rex}_{user}) \\ \mathbf{k}(\mathbf{rex}_{user})^T & k(\mathbf{rex}_{user}, \mathbf{rex}_{user}) \end{bmatrix} \quad (7)$$

$\mathbf{k}(\mathbf{rex}_{user}) = [k(\mathbf{rex}_1, \mathbf{rex}_{user}), \dots, k(\mathbf{rex}_N, \mathbf{rex}_{user})]^T$ 이다. 따라서 식 (6)의 확률밀도함수를 최대화함으로써 저차원 표정 벡터  $\mathbf{rex}_{user}$ 로부터 원하는 고차원 AC 표정 벡터  $\mathbf{daex}$ 를 구할 수 있다.

그림 3은 이상에서 설명한 표정 학습 및 생성에 관련된 SGPLVM 알고리즘을 도식화한 것이다.

SGPLVM 학습을 통해 출력된 표정 생성 파라미터  $\{\mathbf{rex}_i\}, \alpha, \beta, \gamma, \mathbf{w}$ 는 SGPLVM 표정 생성기로 입력되며, 이는 표정 생성 인터페이스 공간에서 사용자가 선택한 표정 벡터  $\mathbf{rex}_{user}$ 에 관련된 원하는 AC 표정 벡터  $\mathbf{daex}$ 를 생성하는데 사용된다. AC 표정 벡터  $\mathbf{daex}$ 는 학습 표정 벡터의 평균  $\mathbf{m}$ 과 더해져서 최종의 원하는 표정 벡터(desired expression), 즉  $\mathbf{dex}$ 를 복원하며, 캐릭터의 얼굴 표정은 벡터 그래픽 드로잉 툴을 통해 사용자에게 보여진다.

### 3. 얼굴 표정 모델의 학습 및 생성

#### 3.1 표정 모델의 학습

본 논문에서는 Russell의 내적 정서 상태에 근거하여 재정의된 벡터 그래픽 캐릭터의 26가지 고차원

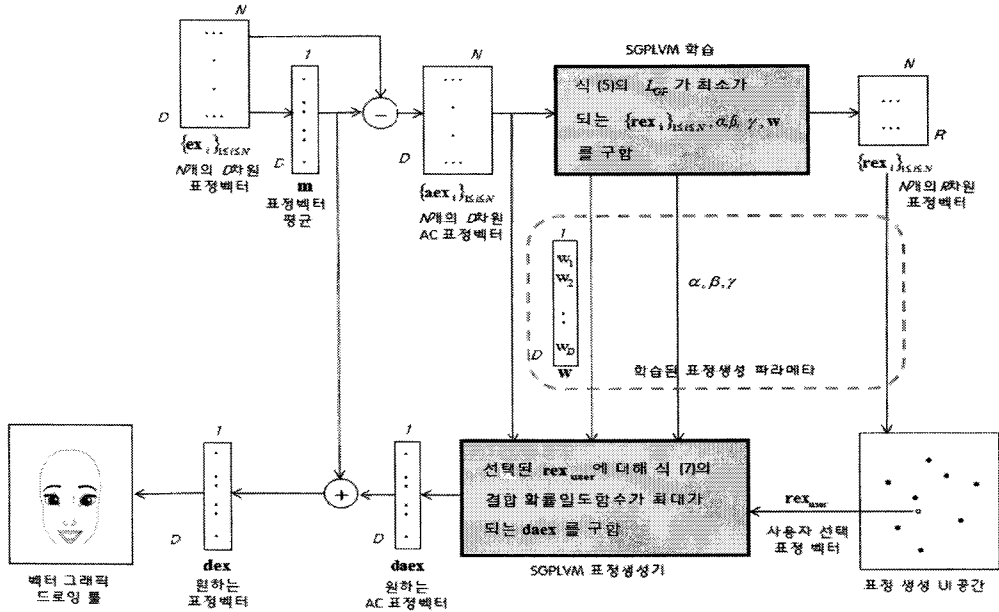


그림 3. 표정 학습 및 생성을 위한 SGPLVM 알고리즘

표정 데이터로부터 SGPLVM 모델의 확률분포함수를 학습하고 이로부터 추정된 저차원의 파라메타들을 사용하여 실시간으로 새로운 표정을 생성한다. 이때 각 캐릭터들의 표정 공간은 스타일에 무관하게 동일하며, 각 스타일에 따른 몇 개의 파라메타만을 조절함으로써 다양한 캐릭터들의 표정을 효율적으로 생성할 수 있다.

### 3.1.1 표정 모델의 학습

미국의 상업용 애니메이션 캐릭터는 많은 대중적 인지도를 얻고 있으며, 얼굴과 성격에 관한 류근영 [19]의 선행 연구에서 “미국 애니메이션 캐릭터의 음향오행 분류에 따른 외적 형태가 내적 성격과 일관되고 보편적인 연관성이 높다”고 평가된 바 있다. 이러한 선행 연구를 바탕으로 고혜영 [12]은 Russell의 내적 정서 상태에 근거하여 미국의 애니메이션 캐릭터의 파장된 표정을 재정의하였다.

본 논문에서는 표정 모델의 학습을 위해 고혜영의 연구에서 재정의한 디즈니 영화 “미녀와 야수”의 등장 인물인 벨과 케스톤의 26가지 표정을 사용하였으며, 그림 4는 대표적인 9가지 표정만을 보인 것이다.

각 캐릭터의 26가지 표정 데이터로부터 추출된 2880 차원의 특징 벡터  $ex$ 는 표정 모델 파라메타를

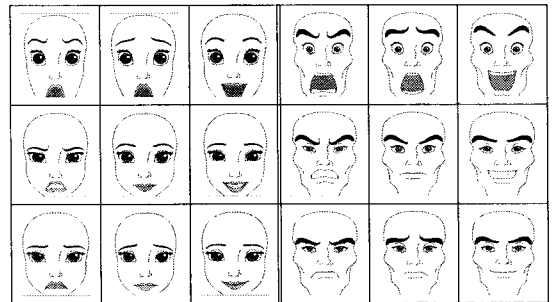


그림 4. 표정 모델의 학습에 사용된 9가지 표정의 벡터 그래픽 캐릭터

학습하기 위해 식 (4)의 부 로그-사후확률에 사용되었으며, 본 논문에서는 고차원의 특징 벡터  $ex$ 로부터 2차원 잠재공간 벡터  $rex$ 를 모델링하였다. 또한 부 로그-사후확률을 최적화하기 위해 Matlab 7.1에서 Nelder-Mead 방법 [20]을 사용하였으며,  $\alpha, \beta, \gamma$ 의 초기값은 1로 설정하였다. 수렴 속도를 향상시키기 위해  $\{rex_i\}$ 의 초기값은  $\{ex_i\}$ 에 대한 2차원 고유벡터 (eigen-vector)로 가정하였다.

그림 5는 각 캐릭터의 26개 표정에 대한 2차원 잠재 표정 공간을 보인 것이다. 그림에서 기호 +는 학습에 사용된 26개 표정을 나타내며, 이 중 몇 개의 표정은 실선으로 연결하였다.

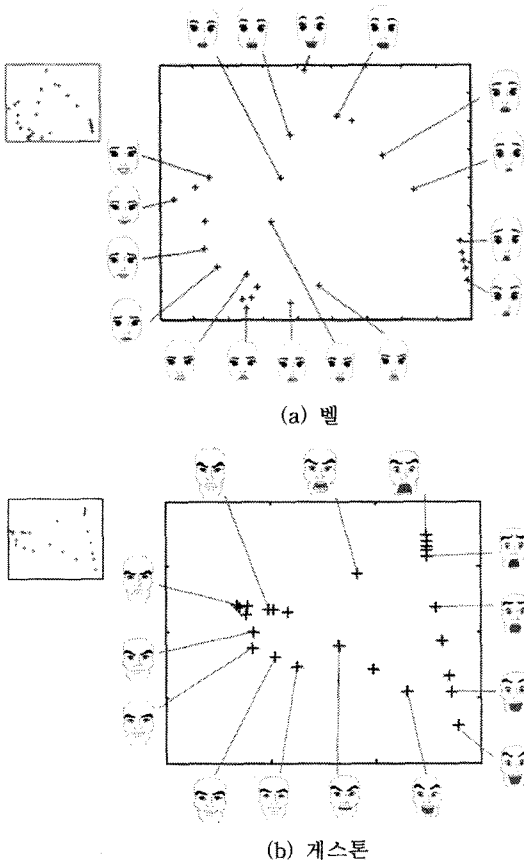


그림 5. 각 캐릭터의 2차원 잠재 표정 공간

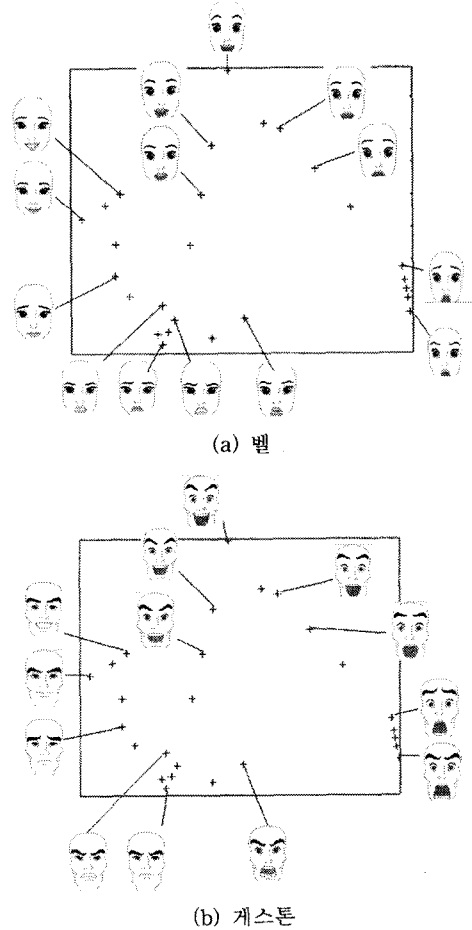


그림 6. 캐릭터들의 고정된 2차원 잠재공간

시뮬레이션 결과로부터 캐릭터의 표정 공간은 스타일에 따라 서로 다르게 나타남을 알 수 있다. 그러나 유사한 표정들은 2차원 공간 상에서 서로 인접하며, 26개의 연속된 표정들의 움직임은 거의 유사하다. 특히 벨 캐릭터의 표정 공간은 게스톤 표정 공간을 상하 대칭한 후 시계 방향으로 45° 회전시킨 것과 유사함을 알 수 있다.

### 3.1.2 표정 공간의 통일

다양한 캐릭터들의 표정 모델을 효율적으로 학습하고 표정 생성 인터페이스의 구현을 단순화하기 위해서는 캐릭터들의 다양한 표정 공간을 하나의 표정 공간으로 고정시킬 필요가 있다.

따라서 본 논문에서는 그림 6과 같이 캐릭터들의 표정 공간을 벨 캐릭터의 잠재공간 상에 고정시킨 후 이에 따라 각 캐릭터의 모델 파라메타  $\alpha, \beta, \gamma, \mathbf{w}$ 를 학습하였다.

2차원 표정 공간  $\{\mathbf{rex}_i\}$ 을 고정시킨 후 파라메타  $\alpha, \beta, \gamma$ 의 학습을 위해 변형한 부 로그-사후확률  $L_{FIX-GP}$ 은 다음과 같다.

$$\begin{aligned}
 L_{FIX-GP} &= -\ln p(\alpha, \beta, \gamma, \mathbf{w} | \{\mathbf{rex}_i\}, \{\mathbf{aex}_i\}) \\
 &= -\ln p(\{\mathbf{rex}_i\}, \alpha, \beta, \gamma, \mathbf{w} | \{\mathbf{aex}_i\}) + \ln p(\{\mathbf{rex}_i\}) \\
 &= \frac{D}{2} \ln |\mathbf{K}| + \frac{1}{2} \sum_{i=1}^D w_i^2 \mathbf{aex}_i^T \mathbf{K}^{-1} \mathbf{aex}_i + \ln \frac{\alpha \beta \gamma}{\prod_{i=1}^D w_i^N}
 \end{aligned} \tag{8}$$

또한 각 표정 벡터에 대한 스케일링 파라메타  $\mathbf{w}$ 는 구해진 파라메타  $\alpha, \beta, \gamma$ 와  $\{\mathbf{rex}_i\}$ 를 다음의 수식에 대입하여 구한다.

$$w_i = \sqrt{N / (\mathbf{aex}_i^T \mathbf{K}^{-1} \mathbf{aex}_i)} \tag{9}$$

따라서 본 논문에서 수정한 표정 모델 학습 알고리즘을 정리하면 다음과 같다.

```

function LEARNMODEL({exi}, {xi})
//{exi}1 ≤ i ≤ N은 캐릭터의 N개의 표정벡터
//{rexi}는 SGPLVM에 의해 이미 학습된
  2차원 표정 벡터
// α=1,β=1,γ=1,w=1로 초기화
for loop=1 to 반복횟수
  for i=1 to N // N개의 표정 벡터에 대해
    α,β,γ에 관해 LFIX-GP를 최소화
  end if
  for 각 표정의 특징 벡터 차원 D에 대해
    wi = √N/(aexiT·K-1·aexi)-1
  end for
end for
return {α, β, γ, w}
    
```

3.2 표정 생성 및 제어

{rex<sub>i</sub>}와 각 캐릭터의 표정 모델 파라메타 α,β,γ,w가 학습되고 나면 사용자 선택의 2차원 표정 벡터 rex<sub>user</sub>에 대한 원하는 고차원 AC 표정 벡터 daex는 다음과 같이 식 (6)에 대한 부 로그-사후확률을 최소화함으로써 구할 수 있다.

$$L_p(\mathbf{rex}_{user}, \mathbf{daex}) = \frac{\|\mathbf{w}(\mathbf{daex} - f(\mathbf{rex}_{user}))\|^2}{2\sigma^2(\mathbf{rex}_{user})} + \frac{D}{2} \ln \sigma^2(\mathbf{rex}_{user}) + \frac{1}{2} \|\mathbf{rex}_{user}\|^2 \tag{10}$$

이때 학습된 SGPLVM 모델 파라메타에 의한 rex<sub>user</sub>의 예측값은 식 (11)과 같다.

$$f(\mathbf{rex}_{user}) = \mathbf{m} + \mathbf{aex}_j^{-1} \cdot \mathbf{K}_j^{-1} \cdot \mathbf{k}(\mathbf{rex}_{user}) \tag{11}$$

또한 σ<sup>2</sup>(rex<sub>user</sub>)는 식 (12)와 같이 모델의 불확실 정도를 나타내며, 학습에 사용된 표정 데이터의 근처에서 작은 값을 가진다.

$$\sigma^2(\mathbf{rex}_{user}) = k(\mathbf{rex}_{user}, \mathbf{rex}_{user}) - \mathbf{k}(\mathbf{rex}_{user})^T \cdot \mathbf{K}_j^{-1} \cdot \mathbf{k}(\mathbf{rex}_{user}) \tag{12}$$

4. 실험 결과 및 고찰

4.1 표정 생성 인터페이스

그림 7은 표정 생성 인터페이스의 화면 구성과 표정 생성의 한 예를 보인 것이다.

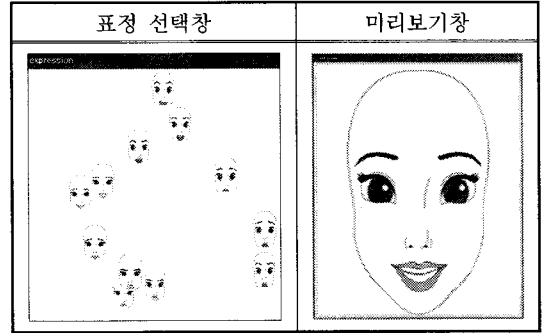


그림 7. 표정 생성 인터페이스의 화면 구성 및 표정 생성

표정 생성 인터페이스는 통일된 캐릭터의 2차원 잠재공간 {rex<sub>i</sub>} 상에서 표정을 선택할 수 있는 표정 선택창과 사용자가 선택한 표정 벡터 rex<sub>user</sub>로부터 예측된 표정을 보여주는 미리보기 창으로 구성된다.

이때 사용자 선택의 표정 벡터 rex<sub>user</sub>에 관한 표정은 식 (11)과 같이 학습된 표정 모델 파라메타 {rex<sub>i</sub>}, α,β,γ,w를 사용함으로써 예측된다.

이와 같이 표정 선택창에서 마우스를 드래그하면 해당 위치의 표정이 미리보기 창에서 실시간 구현된다.

그림 8은 표정 선택창에서 드래그 위치에 따라 생성되는 캐릭터 벨의 다양한 중간 표정들을 보인 것이다.

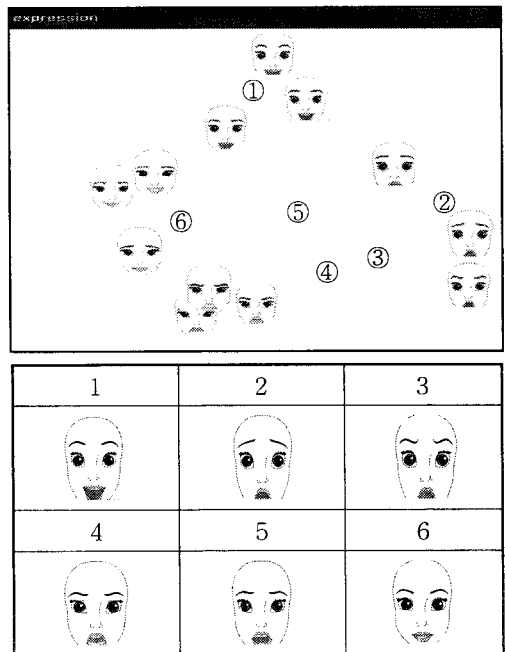
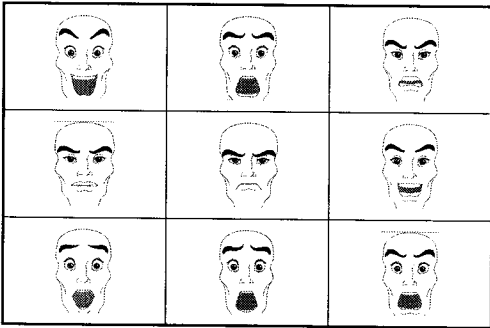


그림 8. 드래그에 따른 표정 생성의 예

캐릭터 게스톤의 경우에도 인터페이스 상에서 마우스를 드래그함으로써 그림 8과 같이 다양한 표정이 생성됨을 알 수 있다.



이와 같이 사용자는 2차원 공간상에서 직관적으로 새로운 표정을 실시간으로 생성하고 제어할 수 있게 된다.

#### 4.2 표정 보간

본 논문에서는 SGPLVM 모델에 의해 생성된 캐릭터들의 보간을 위해 간단한 접근을 시도하였다. 이는 본 논문에서 구현한 캐릭터들의 2차원 표정 공간이 서로 동일하기 때문에 새로운 좌표축  $z$ 를 추가하여 각 캐릭터의 표정 공간을 분리시킨 후, 식 (13)과 같이 블렌딩 인자(blending factor)  $s$ 의 값을 변화시킴으로써 동일한 표정을 가지는 중간 캐릭터를 생성하는 것이다.

$$L(\mathbf{r}_{\text{ex}_{\text{user}}}, \mathbf{d}\mathbf{a}\mathbf{e}\mathbf{x}) = (1-s)L_{p1}(\mathbf{r}_{\text{ex}_{\text{user}}}, \mathbf{d}\mathbf{a}\mathbf{e}\mathbf{x}) + sL_{p2}(\mathbf{r}_{\text{ex}_{\text{user}}}, \mathbf{d}\mathbf{a}\mathbf{e}\mathbf{x}) \quad (13)$$

여기서  $L_{p1}$ 과  $L_{p2}$ 는 각각 벨과 게스톤 캐릭터의 부로그-사후확률이며,  $\mathbf{r}_{\text{ex}_{\text{user}}}$ 는 사용자 선택의 2차원 표정 벡터,  $\mathbf{d}\mathbf{a}\mathbf{e}\mathbf{x}$ 는 이에 관련된 원하는 고차원 AC 표정 벡터이다.  $s$ 는 0과 1 사이의 실수이다.

그림 9는 두 캐릭터의 표정 공간을 3차원 좌표계로 설정하고,  $z=0,1$ 인 좌표상에 각각 벨과 게스톤 캐릭터의 2차원 잠재 표정 공간을 배치시킨 것이다.

시뮬레이션 결과로부터 동일한 표정의 각 캐릭터에 대해  $s$ 의 값을 0에서 1까지 변화시킴으로써 같은 표정의 중간 캐릭터를 생성할 수 있음을 알 수 있다. 이는 축의 확장을 통해 더 많은 캐릭터들의 표정을 보간할 수 있음을 의미한다.

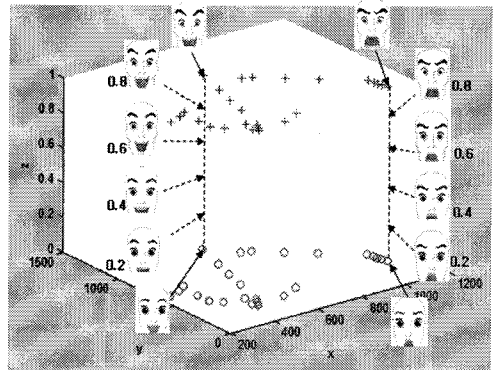


그림 9. 3차원 표정 공간 상에서 두 캐릭터의 표정 보간

### 5. 결론 및 향후과제

본 논문에서는 가우시안 프로세스 모델에 기반하여 애니메이션 캐릭터의 연속된 얼굴 표정을 2차원 표정 공간 상에서 실시간으로 구현하기 위한 자동 표정 생성 시스템을 제안하였다.

표정 학습에 사용된 벡터 그래픽 캐릭터들은 스타일에 따라 서로 다른 표정 공간을 가지나 2차원 표정 공간을 학습하기 위한 최적화 수식을 변형함으로써 고정된 2차원 표정 공간 상에 배치시킨 후 캐릭터에 따라 나머지 학습 파라미터들을 학습하였다. 또한 간단한 보간을 이용하여 동일한 표정을 가지는 중간 캐릭터들을 생성하였다. 이로써 보다 많은 캐릭터들의 표정 모델을 효과적으로 학습할 수 있으며, 표정 생성 인터페이스의 구현을 단순화할 수 있다. 또한 사용자는 표정과 정서와의 관련성에 대한 사전 지식과 관찰 없이도 연속되는 다양한 캐릭터들의 표정을 직관적으로 쉽게 생성할 수 있음을 확인할 수 있었다.

향후 정면 캐릭터 뿐만 아니라 각도를 가진 다양한 캐릭터들의 표정을 분석하고 학습에 사용되지 않은 새로운 표정을 생성할 수 있는 자동 표정 생성 시스템을 구현한다면 캐릭터 애니메이션 구현의 효율성을 높이는 데 기여할 수 있을 것이다.

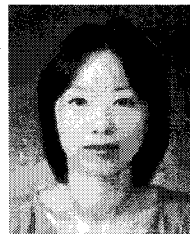
### 참고 문헌

- [1] Frederic I. Parke, Keith Waters, Computer Facial Animation, A. K. Peters, Wellesley, Massachusetts, 1996.
- [2] B. Guenter, C. Grimm, D. Wood, H. Malvar,



- and F. Pighin, "Making Faces," Proceedings of SIGGRAPH 98, pp. 55-66, 1998.
- [3] J.J. Choi, D.S.Kim and I.K.Lee : Anticipation for Facial Animation, In proceeding of Computer Animation and Social Agents (CASA), 2004.
- [4] Pyun, H., Kim, Y., Chae, W., Kang, H. W. & Shin, S. Y., "An example-based approach for facial expression cloning," ACM Transactions on Graphics (SIGGRAPH 2006).
- [5] Plutchik, R. Emotions : A Psycho evolutionary Synthesis, New York ; Harper & Row, 1980.
- [6] P. Ekman, Emotion in the human face, Second Edition. Cambridge University Press, 1982.
- [7] Russell, J. A., A circumplex model of affect. Journal of Personality and Social Psychology, 1980.
- [8] P. Ekman, W. V.Friesen, "Facial Action Coding System," Psychologists Press, 1977.
- [9] F. I. Parke, "Parameterized models for facial animation," IEEE computer Graphics and Applications, 2:61-68, 1982.
- [10] Keith Waters, "A Muscle Model for Animation Three-Dimensional Facial Expression," Computer Graphics, Vol.21, No.4, 1987.
- [11] 강만석, 최종일, "국내 애니메이션 산업 발전방안 연구," 한국방송영상산업진흥원, 2001.
- [12] Khronos Group(차세대 이동통신 포럼) <http://www.khronos.org/opencv/>.
- [13] <http://kr.blog.yahoo.com/feelif2/1740>.
- [14] 고희영, 이재식, 김재호, "내적 정서 상태 차원에 근거한 캐릭터 표정 생성 프로그램 구현," 한국해양정보통신학회논문지, 제12권, 제3호, pp. 438-444, 2008.
- [15] P.C. Yuen and J.H. Lai, "Face Representation Using Independent Component Analysis," Pattern Recognition, Vol.35, No.6, pp. 1247-1257, 2002.
- [16] 박태희, "간소화된 주성분 벡터를 이용한 벡터 그래픽 캐릭터의 얼굴 표정 생성," 한국해양정보통신학회논문지, 제12권, 제9호, pp. 1547-1553, 2008.
- [17] 이해구 저, 포토샵 cs2+일러스트레이터 cs2 : okokok 알찬 예제로 배우는, 교학사, 2007.
- [18] Keith Grochow, Steven Martin, Aaron Hertzmann, and Zoran Popovic, "Style-based inverse kinematics," ACM transactions on Graphics(Proceedings of SIGGRAPH 2004), Vol.23, No.3, pp. 522-531, 2004.
- [19] 류근영, 음양오행(陰陽五行) 사상에 의한 캐릭터의 두상과 체상의 유형 분석 연구, 부산대학교 석사학위논문, 2002.
- [20] 정상권, 이승래, 권옥현 편저, 수치해석, 도서출판 한산, 2000.

**박 태 희**



1993년 부경대학교 정보통신공학과 공학사  
 1996년 부경대학교 전자공학과 공학석사  
 1999년 부산대학교 전자공학과 박사 수료

현재 동명대학교 메카트로닉스공학과 전임강사  
 관심분야 : 비디오영상처리, 영상압축, 얼굴 애니메이션

**김 재 호**



1980년 부산대학교 전기기계공학과 공학사  
 1982년 한국과학기술원 산업전자공학과 공학석사  
 1990년 한국과학기술원 전기및전자공학과 공학박사

현재 부산대학교 전자공학과 정교수  
 관심분야 : 영상처리, 영상압축, VLSI, SOC 설계