일반논문 (Regular Paper)
방송공학회논문지 제24권 제3호, 2019년 5월 (JBE Vol. 24, No. 3, May 2019)
https://doi.org/10.5909/JBE.2019.24.3.440
ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

비디오 상의 얼굴에 대한 3차원 변형 시스템

박 정 식[a], 서 병 국[b], 박 종 일[a]‡

A System for 3D Face Manipulation in Video

Jungsik Park[a], Byung-Kuk Seo[b], and Jong-Il Park[a]‡

요 약

본 논문에서는 비디오 상의 얼굴을 사용자가 원하는 대로 3차원적으로 변형시켜볼 수 있도록 하는 시스템을 제안한다. 제안된 시스템의 3차원 얼굴 변형은 비디오 프레임의 얼굴 영역에 사용자가 변형을 가한 3차원 얼굴 모델을 덮어 씌우는 방식으로서, 기존의 애플리케이션이나 방법과 달리 비디오 상에서 3차원 변형을 실시간으로 가할 수 있도록 한다. 이를 위해 변형 가능한 3차원 얼굴 모델을 영상과 정합하고, 동시에 사용자가 가한 변형을 정합된 모델에 적용, 프레임 영상을 텍스처 매핑하여 렌더링한다. 이러한 과정은 많은 연산을 요하기 때문에 기능별로 소프트웨어 모듈을 나눠 각각의 쓰레드에서 병렬적으로 처리하도록 구현함으로써 실시간 처리가 가능하도록 하였다. 실험 결과를 통해 비디오 상의 얼굴의 눈 주변, 코, 턱, 볼 등 부위들에 대해, 기존 애플리케이션에 비해 자연스러운 변형을 실시간으로 가할 수 있음을 확인할 수 있다.

Abstract

We propose a system that allows three dimensional manipulation of face in video. The 3D face manipulation of the proposed system overlays the 3D face model with the user 's manipulation on the face region of the video frame, and it allows 3D manipulation of the video in real time unlike existing applications or methods. To achieve this feature, first, the 3D morphable face model is registered with the image. At the same time, user's manipulation is applied to the registered model. Finally, the frame image mapped to the model as texture, and the texture-mapped and deformed model is rendered. Since this process requires lots of operations, parallel processing is adopted for real-time processing; the system is divided into modules according to functionalities, and each module runs in parallel on each thread. Experimental results show that specific parts of the face in video can be manipulated in real time.

Keyword : Face manipulation, face editing, video editing, face tracking, mesh deformation

a) 한양대학교 컴퓨터소프트웨어학과(Department of Computer Science, Hanyang University)
b) 한국전자통신연구원(Electronics and Telecommunications Research Institute)
‡ Corresponding Author : 박종일(Jong-Il Park)
E-mail: jipark@hanyang.ac.kr
Tel: +82-2-2220-0368
ORCID: https://orcid.org/0000-0003-1000-4067

※ 이 논문은 2019년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2017-0-01849, 실내외 임의공간 실시간 영상 합성을 위한 핵심 원천기술 및 개발툴킷 개발)
· Manuscript received August 9, 2018; Revised April 24, 2019; Accepted May 13, 2019.

Copyright ⓒ 2016 Korean Institute of Broadcast and Media Engineers. All rights reserved.
"This is an Open-Access article distributed under the terms of the Creative Commons BY-NC-ND (http://creativecommons.org/licenses/by-nc-nd/3.0) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited and not altered."

Ⅰ. 서 론

오늘날, 사람들은 일상적으로 자신의 모습을 사진 또는 비디오로 촬영하고, 이를 다른 사람과 공유한다. 이 과정에서 자신의 모습을 더 예쁘게 혹은, 재미있게 만들기 위해 사진이나 동영상의 얼굴을 편집하곤 한다. 십여 년 전까지는 이러한 편집은 주로 다양하고 복잡한 기능을 가진 PC 상의 사진편집 프로그램을 통해야만 했기 때문에 해당 프로그램에 익숙하지 않은 사람은 사진을 편집하는 것이 어려웠다. 하지만, 최근에는 영상처리 및 컴퓨터 비전 기술의 발전으로 간단한 조작을 통해 누구나 쉽게 사진이나 비디오 상의 자신의 얼굴을 편집할 수 있게 해주는 스마트폰 애플리케이션이 등장하였다[1][2]. 하지만, 이러한 애플리케이션들은 사람의 얼굴에 대한 3차원 기하구조를 고려하지 않고 단순한 영상 왜곡 기법을 통해 얼굴을 변형시켜준다. 따라서 그림 1의 (2), (3)의 붉은 원으로 표시한 부분과 같이 얼굴뿐만 아니라 배경까지 함께 왜곡되는 문제가 있으며, 원하는 형태로 변형시키는 것도 쉽지 않다. 따라서, 보다 더 자연스러운 변형을 위해서는 3차원 영상편집 기술을 필요로 한다.

그림 1. 기존 얼굴 변형 애플리케이션과 제안하는 시스템을 통한 변형 예시: (1) 원본 영상, (2) [1]을 이용한 의한 변형 결과, (3) [2]를 이용한 변형 결과, (4) 제안하는 시스템을 이용한 변형 결과

Fig. 1. Examples of face manipulation via the existing applications and the proposed system: (1) original image, (2) manipulation result using [1], (3) manipulation result using [2], (4) manipulation result using the proposed system

컴퓨터 그래픽스 분야에서는 3차원 영상편집에 대한 여러 연구가 진행되어 왔다[3-8]. 3차원 영상 편집은 피사체에 대한 3차원 모델을 획득하고, 이를 영상이나 비디오 프레임에 정합한 뒤, 3차원 기하변환이나 변형 등의 편집을 적용하여 렌더링하는 방식으로 이루어진다. 피사체에 대한 3차원 모델의 준비는 사전에 다시점 영상기반 모델링 방법 등을 이용하여 획득하거나[3-4], 편집 대상이 되는 한 장의 영상으로부터 대칭성 등의 제약조건에 기반한 인터랙티브 모델링 방법을 이용하여 생성하거나[5-6], 온라인 모델 데이터베이스로부터 획득한 뒤 영상에 맞춰 조정하는 방식으로[7-8] 이루어진다. 편집 방식 또한, 간단한 복제, 기하변환, 등방성 및 비등방성 스케일링에서부터[5-8], 임의의 형태로의 변형[4-5], 물리기반 변형[3][8]과 같이 다양하게 구현되었다. 이러한 연구들은 피사체의 3차원 기하구조를 반영한 편집이 가능하며 배경을 왜곡시키지 않는다. 하지만, 표정 등에 의해 변화하는 사람의 얼굴을 강체나 간단한 물리 모델로 모델링하기 어렵기 때문에 앞의 방법들을 적용하기 어렵다.

한편, 피사체를 사람의 얼굴로 한정한 정합, 모델링, 편집 기술들 또한 연구되어 왔다. 이러한 기술들에는 다른 얼굴로 표정을 전이시키거나[9], 다른 얼굴로 교체하거나[10], 특정 부위를 변형시키거나[11], 디지털 아바타를 생성하고 표정을 합성하는 방법[12-13] 등이 있으며, 3차원 얼굴 모델과 영상의 정합에 기반한다. 이 중에서 얼굴의 특정 부위를 변형시켜 보여주는 연구[11]는 비록 뛰어난 변형 품질을 보여주었지만, 한 장의 영상에 적용 가능하고, 얼굴 모델의 형태계수의 조절을 통해 변형하기 때문에 변형 가능한 범위가 얼굴 모델링에 사용된 얼굴 데이터에 의존적인 한계점이 있었다. 그리고, 디지털 아바타의 경우[13], 한 장의 영상만으로도 피사체와 유사한 아바타를 생성하여 조명 환경과

표정의 디테일을 반영한 실감 있는 애니메이션을 렌더링할 수 있었다. 하지만, 아바타의 얼굴 변화는 사람이 표현 가능한 표정으로 한정되어 임의의 형태나 과장된 형태는 표현하기 어려웠다.

우리는 이러한 제약 없이, 사용자가 원하는 대로 비디오 상의 얼굴을 3차원적으로 변형이 가능한 시스템을 제안한다. 제안하는 시스템에서는 얼굴 특징 추적 방법과 얼굴에 대한 변형 가능한 3차원 모델(3DMM: 3D Morphable Model)의 피팅 방법을 이용하여 3차원 얼굴 모델을 영상에 정합하고, 동시에 3차원 메쉬 변형 방법을 이용하여 사용자 입력에 따라 얼굴 모델을 변형시킨다. 모델에 피팅에 의한 변형과 사용자에 의한 변형을 모두 반영한 뒤, 영상 프레임을 텍스처 매핑하여 변형된 모델을 프레임의 얼굴 영역에 렌더링한다. 따라서, 기존 애플리케이션이나 연구와 달리, 그림 1의 (4)와 같이 배경에 왜곡이 발생하지 않고, 3차원적으로 사용자가 원하는 대로 얼굴의 변형이 가능하며, 비디오에 적용 가능하기 때문에 변형된 모습을 여러 시점에서 살펴볼 수 있다는 장점을 가진다. 본 논문은 우리의 이전 연구[14]를 확장한 것으로, 얼굴 모델 피팅 방법의 고속화 및 병렬 처리 구현을 통해 실시간으로 동작이 가능하도록 하였다.

Ⅱ. 비디오 상의 얼굴 변형 시스템

본 논문에서 제안하는 시스템에서는 비디오 프레임 영상 내의 얼굴 상태(위치, 자세, 표정 등)에 맞춰진 3차원 얼굴

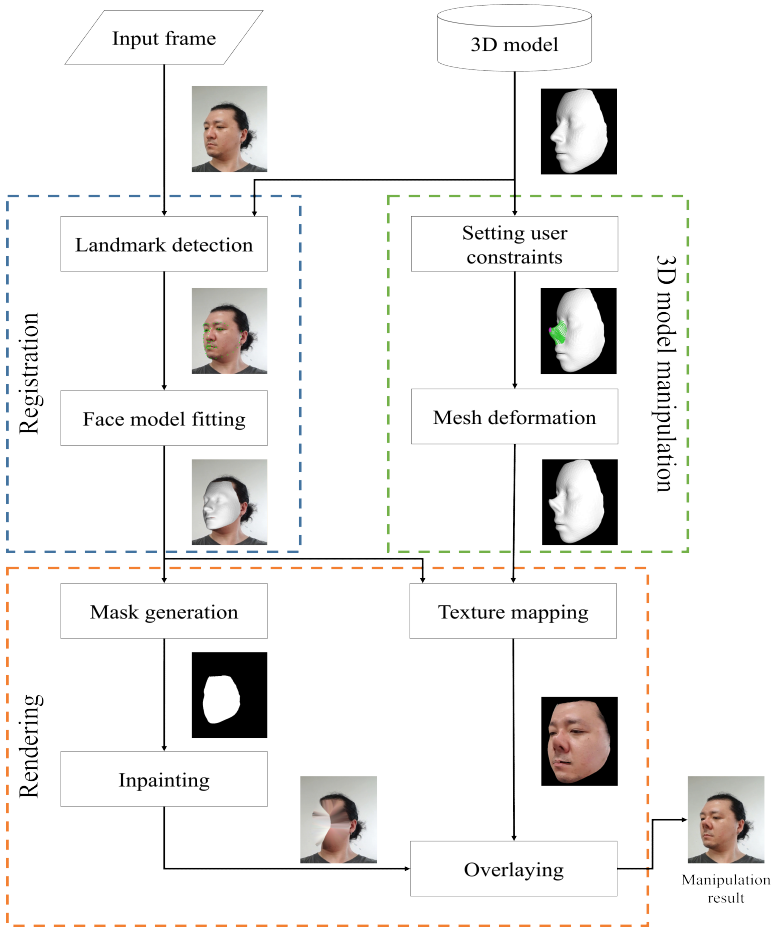

그림 2. 제안하는 얼굴 변형 시스템의 흐름
Fig. 2. Flow of the proposed face manipulation system

모델에 추가적으로 사용자에 의한 변형을 가한 뒤, 영상 내의 얼굴 영역 위에 덮어씌우는 방식으로 얼굴이 3차원적으로 변형된 영상을 생성한다. 이러한 처리 과정은 기능별로 나뉘어 정합(Registration), 변형(Manipulation), 렌더링(Rendering) 소프트웨어 모듈에서 각각 수행되며, 전체적인 흐름은 그림 2와 같다.

1. 정합 모듈

얼굴 모델을 영상에 정합하는 과정은 다시 얼굴 특징 검출과 얼굴 모델 피팅 과정으로 나뉜다. 얼굴 특정 검출은 얼굴 모델을 영상 내의 얼굴 상태에 맞추어 변형하는 데 기준점 역할을 하는 눈, 코, 입, 외곽선과 같은 랜드마크(Landmark)를 찾는 과정이다. 영상에서 랜드마크를 검출하는 방법에는 여러 가지가 있으나, 최근에는 주로 랜덤 회귀 포레스트 방식[15][16]과 심층학습(Deep learning) 기반의 방식[17]이 널리 사용되고 있다. 랜덤 회귀 포레스트는 속도가 매우 빠르다는 장점을 가지지만, 초기 예측에 의존적이기 때문에 초기 예측 오류로 인해 오검출로 이어지는 경우가 종종 발생한다. 반면에, 심층학습 기반 방법은 속도는 느리지만 강건하다는 장점이 있다.

3차원 얼굴 변형에서는 모델의 정확한 정합이 중요하기 때문에, 얼굴 포즈가 변화하더라도 랜드마크의 위치를 최대한 정확하고 강건하게 검출할 수 있는 방법이 더 적합하다고 볼 수 있다. 따라서 제안하는 시스템에서는 속도는 조금 느리더라도 Kowalski 등이 제안한 심층학습 기반의 방법[17]을 통해 랜드마크를 검출한다. 이 방법은 여러 개의 컨벌루션 신경망(Convolutional neural network)을 직렬로 연결한 다단 심층 신경망(Multistage deep neural network)을 이용한 것이다. 각 단계에서 기준 얼굴 자세와의 기하변환 관계, 특징지도, 랜드마크가 존재할 확률에 대한 열지도가 계산되며, 이로부터 랜드마크 위치가 도출된다. 이러한 정보들은 여러 단계의 신경망을 거치며 최적화된다.

얼굴 특징 검출을 통해 얼굴의 랜드마크의 좌표를 구하게 되면, 이를 이용하여 머리의 포즈를 계산하고 얼굴에 대한 3차원 모델을 영상에 피팅시킴으로써 얼굴과 모델을 정합한다. 얼굴의 형태는 사람마다 제각각이고 표정에 따라 변화하기 때문에, 특정되지 않은 사람의 얼굴을 비디오 상에서 변형하는데 있어서 변형 가능한 3차원 모델이 효과적이다. 제안하는 시스템에서 얼굴에 대한 모델과 피팅 방법은 Huber 등이 제안한 것을 속도 측면에서 개량하여 적용하였다[18]. 변형 가능한 얼굴 모델은, 여러 명의 얼굴에 대한 3차원 스캔 데이터로부터 중립(eutral) 표정의 평균 얼굴 $\overline{\mathbf{M}}$과 이를 기준으로 주성분분석(Principal component analysis, PCA)을 통해 정의된 형태에 대한 기저 $\mathbf{M}_{sb}$ 및 표정에 대한 기저 $\mathbf{M}_{eb}$로 표현된다:

$$\mathbf{M}=\overline{\mathbf{M}}+\mathbf{M}_{sb}\boldsymbol{\alpha}+\mathbf{M}_{eb}\boldsymbol{\beta}. \tag{1}$$

이 모델의 중립 얼굴에 랜드마크 좌표를 사전에 매핑해두고, 얼굴 특징 검출 결과와 매핑된 3차원 좌표를 이용하여 머리의 포즈를 계산한다. 머리의 포즈가 계산되면, 투영된 모델에 매핑된 랜드마크와 영상에서 검출된 랜드마크로 정의되는 다음의 에너지 함수를 최소화함으로써 변형 가능한 모델의 주성분 기저에 대한 형태 계수 $\boldsymbol{\alpha}$를 구할 수 있다:

$$\tilde{\boldsymbol{\alpha}} = \operatorname{argmin}_{\boldsymbol{\alpha}}\left(\sum_{i=1}^{N}\frac{(\mathbf{P}\mathbf{X}_i(\boldsymbol{\alpha})-\mathbf{x}_i)^2}{2\sigma^2} + \lambda \| \boldsymbol{\alpha} \|_2^2\right). \tag{2}$$

여기서, N은 랜드마크의 개수, σ는 랜드마크 좌표에 대한 표준편차, $\mathbf{P}$는 카메라 투영 행렬, $\mathbf{X}_i(\alpha)$와 $\mathbf{x}_i$는 각각 모델에 매핑된 i번째 랜드마크의 3차원 좌표와, 이에 대응되는 검출된 랜드마크의 2차원 좌표를 나타내며 λ는 정규화(regularization) 파라미터이다. 모델의 3차원 랜드마크 좌표 $\mathbf{X}(\alpha)$는 얼굴 형태에 대한 주성분 기저 행렬 $\mathbf{M}_{sb}$와 계수의 선형 결합을 평균 얼굴 $\overline{\mathbf{M}}$와 합함으로써 다음과 같이 결정된다.

$$\begin{aligned}&\mathbf{X}(\boldsymbol{\alpha}) \subset \mathbf{M}_s \\ &\mathbf{M}_s = \overline{\mathbf{M}} + \mathbf{M}_{sb}\boldsymbol{\alpha} .\end{aligned} \tag{3}$$

표정 계수 $\boldsymbol{\beta}$는 추정된 얼굴 형태 $\mathbf{M}_s$로부터 비슷한 방식으로 계산된다. 이러한 과정을 반복적으로 수행함으로써 얼굴 포즈와 주성분 기저에 대한 형태 계수를 최적화한다.

하지만, 그림 3과 같이 얼굴의 윤곽 부분에 해당하는 랜드마크는 위치가 일정하지 않고, 얼굴 포즈에 따라 위치가

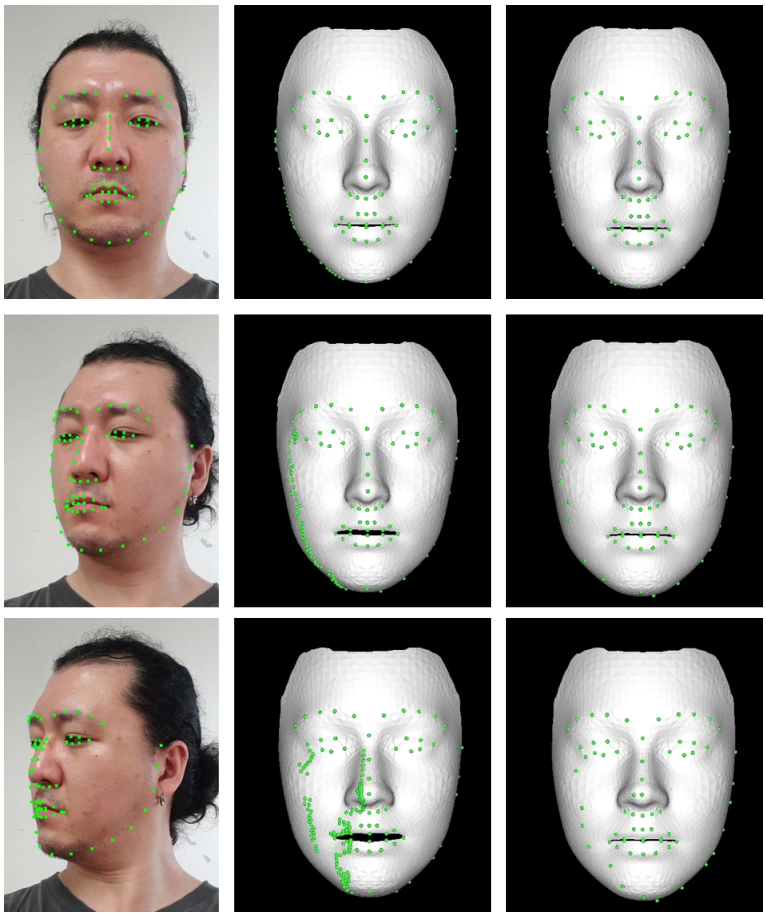

그림 3. 윤곽 랜드마크에 대한 대응점 탐색 결과: (좌) 영상에서 검출된 랜드마크, (중앙) [18]의 방법에 따라 탐색된 대응점, (우) 제안된 시스템에서 수정된 방법에 따라 탐색된 대응점

Fig. 3. Correspondence searching results for contour landmarks: (left) landmarks detected from image, (middle) correspondences detected using the method in [18], (right) correspondences detected using the modified method in the proposed system

달라지게 된다. 따라서, 해당 랜드마크에 대해서는 사전에 매핑된 랜드마크와 모델 간의 관계를 그대로 사용할 수 없고 대응점을 탐색하는 과정이 필요하다. 대응점 탐색 과정은 모델의 정점 중에서 현재 포즈에서 볼 때 전면에 속하면서 다른 정점을 가리고 있는 정점을 골라 영상 평면에 투영시킨 뒤, 투영된 각 점의 좌표와 얼굴 윤곽에서 검출된 랜드마크 간의 거리가 일정 값 이하인 것을 대응점으로 선택하는 방식이다. 하지만, 이런 경우 랜드마크 하나에 여러 개의 3차원 점이 대응되기 때문에 연산량이 많아지게 되고, 특히, 얼굴이 옆으로 돌아가는 경우 속도 저하가 두드러져 실시간 처리가 어려워지는 문제가 있다(그림 3, 가운데 열).

제안된 시스템에서는 이러한 문제를 해결하기 위해 대응점을 한 번 더 필터링하고 보간하는 과정을 추가하였다. 한 랜드마크에 대응점이 여러 개가 선택되는 경우 가장 거리가 가까운 점을 3개까지만 선택하고, 이들이 이루는 메쉬와 랜드마크를 역투영한 광선과의 교점을 계산함으로써 대응점을 하나만 추려낸다(그림 3, 오른쪽 열).

2. 변형 모듈

얼굴 모델이 영상에 피팅된 이후 사용자 입력에 따라 모델을 변형하게 되는데, 이는 As-rigid-as-possible (ARAP) 메쉬 변형 방법을 통해 이루어진다[19]. ARAP는 메쉬의 국소 표면에 대한 회전변환의 적용, 두 단계의 최적화를 통해

국소 표면의 강성(rigidity)을 유지하면서 자연스러운 변형이 가능하여 널리 사용되는 메쉬 변형 방법이다. 메쉬 모델 $\mathbf{M}$이 n개의 정점을 가질 때, 각 정점 $\mathbf{X}_i$, $i \in \{1,\ldots,n\}$과 그 이웃 정점 $\mathbf{X}_j \in N(\mathbf{X}_i)$으로부터 변형에 따른 에너지는 다음과 같이 정의될 수 있다.

$$E = \sum_{\mathbf{X}_i \in \mathbf{M}} \sum_{\mathbf{X}_j \in N(\mathbf{X}_i)} w_{ij} \| (\mathbf{X}_i' - \mathbf{X}_j') - \mathbf{R}_i(\mathbf{X}_i - \mathbf{X}_j) \|^2. \quad (4)$$

여기서, $\mathbf{R}_i$는 정점 $\mathbf{X}_i$에서의 국소 회전변환 행렬, $\mathbf{X}_i'$와 $\mathbf{X}_j'$는 각각 $\mathbf{X}_i$와 $\mathbf{X}_j$의 변형된 좌표를 나타낸다. 변형 에너지의 식의 각 항은 두 단계의 최적화를 통해 최소화된다. 첫 번째 단계에서는 정점들의 위치를 고정시킨 상태에서 국소 회전변환 행렬을 구하고, 두 번째 단계에서는 국소 회전변환 행렬을 고정시킨 상태에서 각 정점들이 사용자 제약조건을 만족시키면서 에너지를 최소화하기 위해 이동되어야 할 좌표를 계산하게 된다.

변형 에너지를 최소화하는 과정에서 제어 정점의 좌표가 제약조건으로 사용되며, 이는 보통 사용자에 의해 지정된다. 즉, 사용자가 지정한 제어 정점을 이동시킨 후 메쉬 변형을 수행하면, 제어 정점의 위치는 이동시킨 위치로 고정된 채로, 그 외의 주변 정점의 좌표가 변형 에너지를 최소화하도록 이동된다. 그림 4는 사용자 제약조건을 설정하고 이를 이동시켜 변형을 수행하는 예시를 보여준다. 원본 모델(좌)에 대해 제어 정점(중앙, 보라색 표시)과 변형될 정점(중앙, 녹색 표시)를 설정한 뒤, 제어 정점을 이동시켜 변형된 결과(우)이다.

3. 렌더링 모듈

렌더링 모듈에서는 비디오 프레임 영상과 변형된 얼굴 모델을 합성하여 변형된 얼굴 영상을 생성한다. 얼굴이 변형됨에 따라 영상 내의 얼굴 영역이 부분적으로 축소하는 경우가 발생할 수 있는데, 이 경우에 얼굴에 의해 가려져있던 부분을 채워주기 위해 인페인팅을 적용한다. 먼저, 인페인팅에서 사용할 마스크는 얼굴 모델을 영상 평면으로부터 투영하여 구한 다음, 마스크의 가장자리에서부터 주변 화소를 참조하여 채워나가는 확산 기반의 방법을 적용한다 [20]. 이 방법은 속도가 빠르지만, 채울 영역이 넓거나 질감이 있는 경우에는 자연스럽게 채우기 어렵다는 단점이 있다. 제안된 시스템에서는 머리나 목 등을 제외한 얼굴에 대해 자연스러운 변형을 가하는 것을 전제로 하기 때문에, 이를 고려하면 실제로 채워져야 할 영역은 그리 크지 않다고 볼 수 있다. 따라서, 이러한 점과 실시간성을 고려하여 확산 기반의 방법을 적용하였다.

인페인팅된 영상을 렌더링한 후에는 변형된 얼굴을 얼굴 포즈를 이용하여 영상의 얼굴 위치에 겹쳐 렌더링한다. 먼저, 모델과 영상간의 투영 관계를 이용하여 현재 프레임 영상으로 얼굴의 텍스처 맵을 갱신한 뒤, 변형 상태를 갱신하여 렌더링한다. 이 때, 모델은 정합 모듈의 얼굴 모델 피팅과 변형 모듈에서의 사용자에 의한 변형으로 인한 각기 다른 두 변형 상태를 가지므로 이를 조합한다. 두 변형된 상태를 모두 적용한 렌더링용 모델 $\mathbf{M}'$은 변형 모듈에서의 변형된 상태를 $\mathbf{M}_m$라 할 때 식 (1)로부터 다음과 같이 된다.

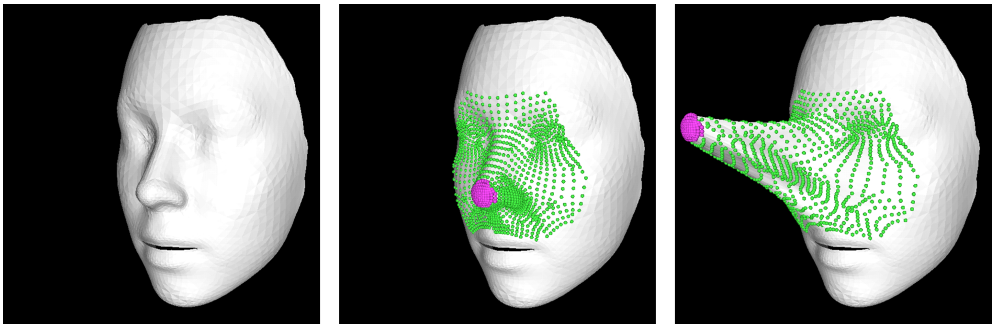

그림 4. 사용자 제약조건(제어 정점: 보라색 점, 변형될 정점: 녹색 점)의 설정과 메쉬 변형의 예시: (좌) 원본 메쉬 모델, (중앙) 사용자 제약조건의 설정, (우) 제어 정점의 이동에 따라 변형된 메쉬

Fig. 4. An example of setting user constraints (control vertices: pupple points, roi vertices: green points) and mesh deformation: (Left) original mesh model, (middle) setting user constraints, (right) deformed mesh according to the movement of the control vertices

$$\mathbf{M}' = \left(\mathbf{M}_{sb}\tilde{\boldsymbol{\alpha}} + \mathbf{M}_{eb}\tilde{\boldsymbol{\beta}} + \overline{\mathbf{M}}\right) + \mathbf{W}\left(\mathbf{M}_m - \overline{\mathbf{M}}\right). \quad (5)$$

여기서 $\mathbf{W}$는 모델의 각 요소의 변형 정도를 조절하는 가중치 행렬인데, 본 논문에서는 가중치를 모두 1로 두어 사용자에 의한 변형을 그대로 적용하였다. 즉, 얼굴 모델 피팅 결과에 사용자에 의해 변형된 모델과 원본 모델(평균 얼굴) 간의 차이를 합하여 렌더링용 모델을 결정한다.

Ⅲ. 구현 및 실험 결과

제안된 시스템은 전체적으로는 파이썬 애플리케이션으로 구현되었다. 이는 얼굴 특징 검출에 사용된 딥러닝 프레임워크 방법이 Theano로 구현되었기 때문이다. 그 밖의 얼굴 모델 피팅, 메쉬 변형 등의 방법은 C++로 구현되었고, 파이썬에 함수 형태로 바인딩되어 호출되는 방식으로 동작된다. 그리고, 얼굴 모델 피팅 과정에서 연산량이 많으면서 병렬화가 가능한 부분을 OpenMP를 이용하여 병렬화함으로써 수행시간을 단축시켰다. 이러한 부분의 예로는 윤곽 랜드마크의 대응점 탐색을 위해 모델의 모든 정점에 대해 가시성 및 가려짐 여부 검사를 수행하는 부분이다. 표 1은 Intel i7-3770 CPU, NVidia GeForce GTX TITAN GPU가 탑재된 PC에서 3448개의 정점과 6736개의 삼각형으로 이루어진 얼굴 모델을 480p 영상에 피팅하는데 소요되는 시간을 측정한 결과이다.

정면 얼굴과 옆으로 돌아간 얼굴에 대해 두 가지 고속화 처리, 즉, OpenMP의 적용 여부와 2장 1절의 마지막 문단에서 서술한 대응점 필터링 적용 여부를 달리하며 시간을 측정하였다. 두 고속화 처리가 얼굴 모델 피팅 과정의 전체에 걸쳐 적용되거나 영향을 미치는 것은 아니기 때문에, 고속화 처리 적용 여부에 따른 시간 측정 결과가 비례관계로 나타나지는 않는다. 정면 얼굴에 대해서는 소요시간 감소 폭이 그리 크지는 않지만, OpenMP 적용으로 약 10 ms, 대응점 필터링 적용으로 약 5 ms 시간이 단축되었다. 하지만, 얼굴이 옆으로 돌아간 경우에 대해, OpenMP 적용으로 약 37 ms, 대응점 필터링 적용으로 약 29 ms 단축되어 소요시간 감소 폭이 크게 나타나며 실시간으로 처리 가능하게 되었다.

비록, 얼굴 피팅 과정에서 소요시간을 단축시켜서 옆으로 돌아간 얼굴에 대해서도 실시간 처리가 가능하게 되었더라도, 얼굴 피팅 뿐만 아니라, 얼굴 특징 검출, 메쉬 변형, 인페인팅을 포함한 렌더링 등 여러 처리를 요하는 전체 시스템은 실시간으로 동작하기 어려울 수 있다. 표 2를 보면, 정합 모듈, 변형 모듈, 렌더링 모듈의 처리를 순차적으로 수행한다고 가정할 때, 한 프레임에 대한 처리 소요 시간은 약 111 ms로, 9 fps의 속도가 된다. 게다가, 사용자가 제어 정점을 움직이고 있는 동안에는 한 프레임 내에서 여러 번의 메쉬 변형 처리가 수행될 수도 있기 때문에 속도는 더욱 저하될 수 있다. 이러한 문제를 해결하기 위해 다중쓰레드를 이용

표 1. 고속화 여부에 따른 얼굴 모델 피팅에 소요되는 시간
Table 1. The processing time for face model fitting w/o and w/ acceleration

	Without correspondence filtering		With correspondence filtering	
	Without OpenMP	With OpenMP	Without OpenMP	With OpenMP
Frontal face	35.70 ms	25.77 ms	30.15 ms	20.63 ms
Oblique face	97.15 ms	60.31 ms	68.03 ms	30.58 ms

표 2. 처리 단계별 소요 시간
Table 2. Processing time for each process

Module	Process	Time	Subtotal
Registration module	Face alignment	36.28 ms	58.49 ms
	Face model fitting	22.21 ms	
Manipulation module	Mesh deformation	7.14 ms	7.14 ms
Rendering module	Inpainting	31.16 ms	45.62 ms
	Texture mapping and deformation transfer	5.34 ms	
	Rendering	9.12 ms	
Total		111.25 ms	

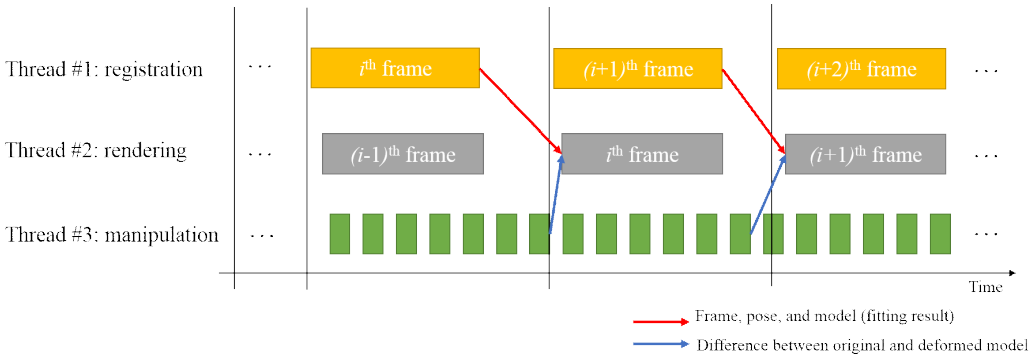

그림 5. 제안된 시스템의 다중쓰레드
Fig. 5. Multithreading in the proposed system

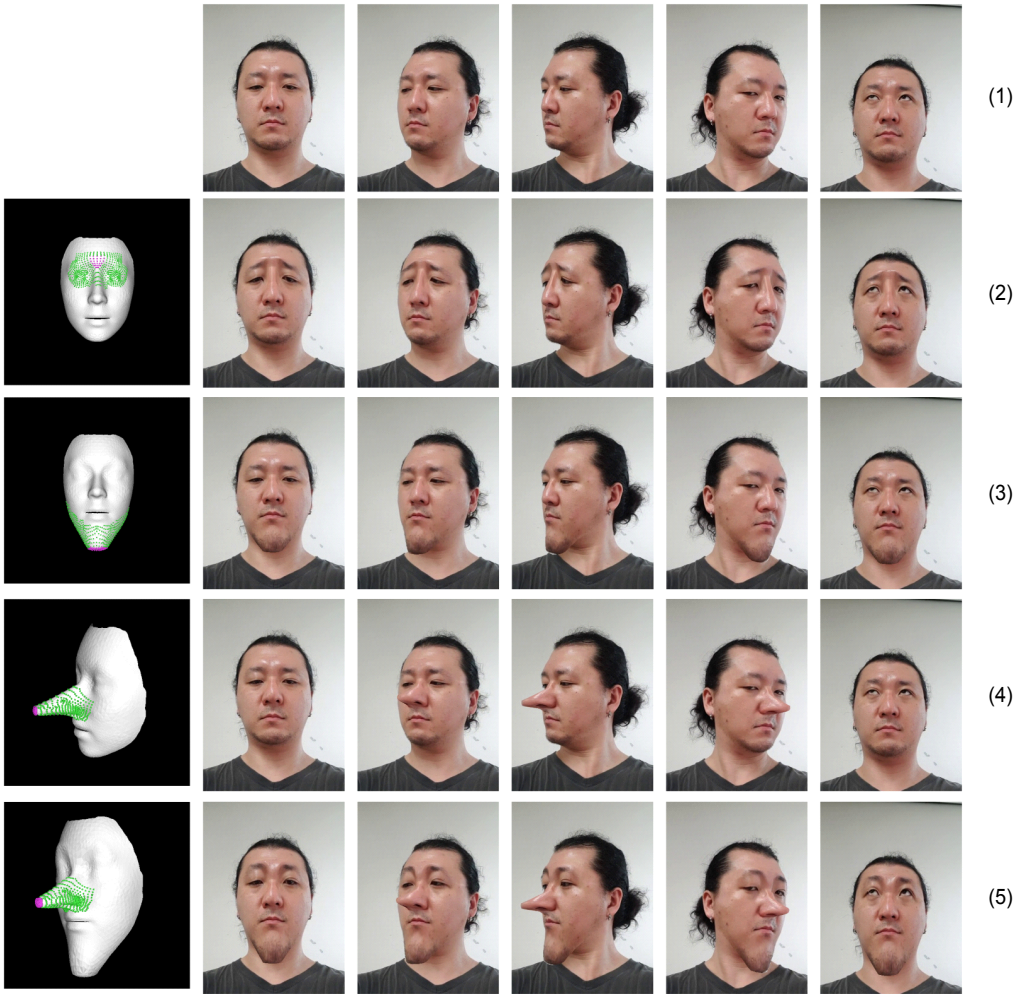

그림 6. 비디오 프레임 대한 얼굴 변형 결과: (1) 원본 프레임, (2)~(4) 눈 주변, 턱, 코를 변형한 결과, (5) (2)~(4)의 변형을 모두 적용한 결과
Fig. 6. Results of face manipulation for video frames: (1) original frames, (2)~(4) results of manipulating the area around eyes, jaw, and nose, (5) results of applying all the manipulations from (2) to (4)

하여 각각의 모듈을 별도의 쓰레드에서 수행하도록 하였다. 이 때, 변형 모듈은 사용자 입력에 따라 프레임과 상관없이 비동기적으로 수행되기 때문에 문제가 없지만, 렌더링 모듈은 얼굴 정합 결과를 이용하여야 하기 때문에 의존성이 생긴다. 따라서 렌더링 쓰레드에서는 현재 프레임이 아닌, 이전 프레임의 정보를 이용하여 렌더링하도록 하였기 때문에 전체 시스템은 최고 약 15 fps의 속도로 동작할 수 있다. 이는 15 fps의 동영상에 대해서는 실시간으로 동작 가능함을 의미한다. 그림 5는 이러한 다중쓰레드 처리를 도식화한 예를 보여준다.

같은 부분에 그림 6는 제안된 시스템을 통해 비디오 프레임(1)의 원본 얼굴에 대해 얼굴 변형을 수행한 결과(2~5)를 보여준다. 첫째 열과 같이 모델에 대해 변형시켰을 때, 2~5번째 열과 같이 변형되며, 이를 통해 배경(얼굴 모델이 커버하지 않는 영역)의 왜곡 없이 얼굴의 눈 주변, 턱, 코에 대한 3차원 변형이 가능함을 볼 수 있다. 하지만, 랜드마크의 좌표 및 얼굴 모델 피팅의 오차로 인해 변형된 부분에 배경의 텍스처가 섞이는 현상과 변형으로 인해 드러나는 가려짐 영역이 커지는 경우 인페인팅 오류가 발생할 수 있기 때문에 얼굴 정합의 정확도 및 인페인팅 품질의 향상이 필요하다.

그림 7은 제안된 시스템을 이용한 변형 결과(세 번째 행)와, 이를 기존의 애플리케이션 중 하나인 [1]을 통해 최대한 비슷하게 재현한 것(두 번째 행)이다. 얼굴에 대한 3차원 모델을 변형시키는 방식의 제안된 시스템과 달리 [1]은 영상의 특정 지점과 일정 반경의 그 주변 영역을 함께 늘이거나 뭉개는 방식이다. 따라서 정밀한 변형이 어렵고, 그림의 늘어난 코 부분과 같이 변형된 부분에 아티팩트가 발생하거나, 목과 티셔츠 부분에서와 같이 배경도 함께 왜곡되기

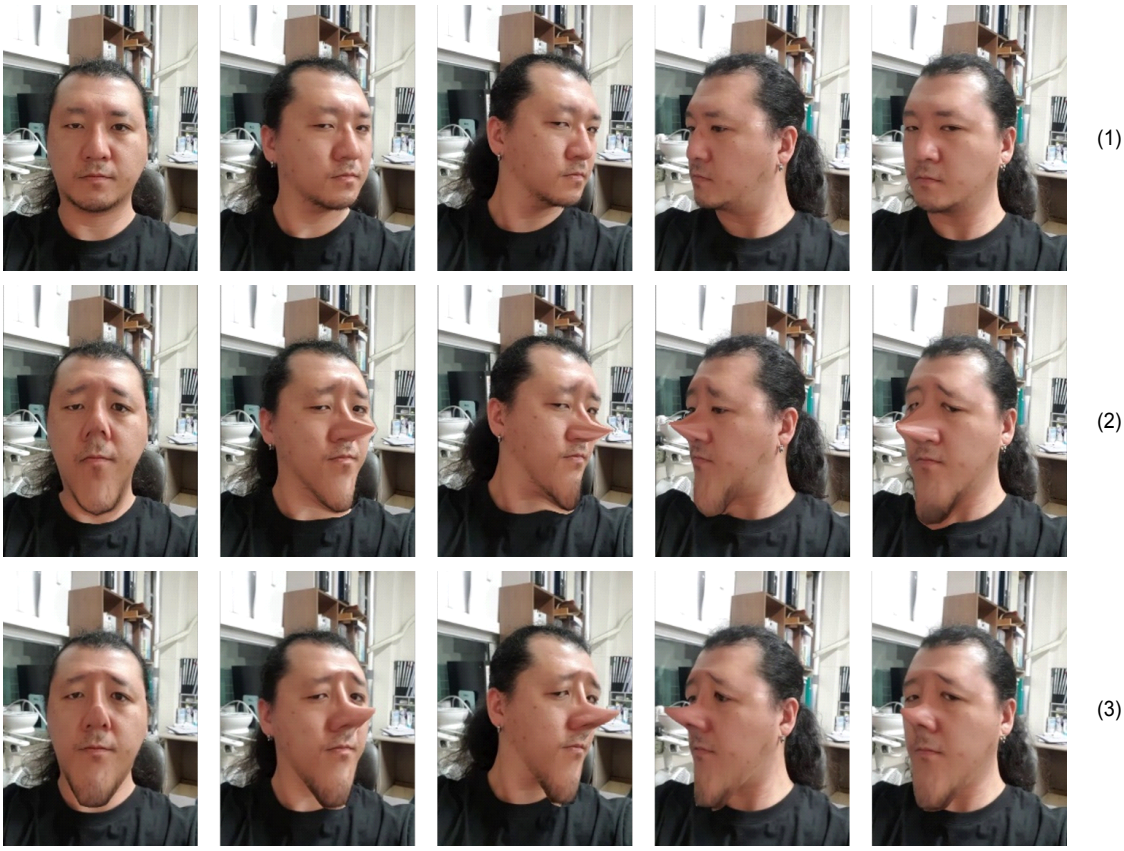

그림 7. 기존 애플리케이션과의 얼굴 변형 비교: (1) 원본 영상, (2) [1]으로 변형한 결과, (3) 제안된 시스템으로 변형한 결과
Fig. 7. Comparison of face manipulation results from the existing application : (1) original images, (2) results from [1], (3) results from the proposed system

그림 8. 기존 애플리케이션과의 얼굴 변형 비교: (1) [2]로 변형한 결과, (2) 제안된 시스템으로 변형한 결과
Fig. 8. Comparison of face manipulation results from the existing application : (1) results from [2], (2) results from the proposed system

도 한다. 또한, 제안된 시스템은 변형된 얼굴 모델을 얼굴 포즈를 이용하여 영상의 얼굴 위치에 렌더링하기 때문에 한 번의 변형으로 여러 프레임에 동일하게 적용 가능하지만, [1]의 경우는 매 프레임 영상을 직접 수정해야하기 때문에 비디오에 적용하기 어렵다는 단점도 있다.

그림 8은 기존 애플리케이션 [2]를 통해 변형된 것(첫 번째 행)과 유사한 형태로 제안된 시스템을 통해 변형을 가한 결과(두 번째 행)를 보여준다. 제안된 시스템과 달리, [2]는 얼굴의 변형과 함께 배경의 왜곡을 수반하여, 정면 얼굴에 대해서는 자연스러운 변형이 가능하지만 얼굴 포즈가 변화할 경우(얼굴을 옆으로 돌리거나 앞으로 숙이는 경우) 부자연스러운 결과를 나타낸다. 또한, 사용자가 원하는 대로 변형이 가능한 제안된 시스템과 달리, [2]는 사전에 정의된 몇 가지 변형 형태만 선택하여 적용할 수 있다는 단점도 있다.

Ⅳ. 결 론

본 논문에서는 비디오 상의 얼굴에 대한 3차원 변형을 시킬 수 있는 시스템을 제안하였다. 얼굴 특징 검출과 변형 가능한 3차원 얼굴 모델의 피팅을 통해 정합된 모델 상태와 메쉬 변형 방법에 의한 변형된 모델 상태를 모두 반영하여 영상 위에 렌더링함으로써 사용자가 원하는 대로 3차원적인 변형이 가능하도록 할 수 있다. 이러한 처리를 통해 기존 애플리케이션에 비해 배경의 왜곡이 발생하지 않고 더 자연스러운 편집이 가능함을 실험을 통해 확인할 수 있었다. 또한, 제안된 시스템은 연산이 많이 소요되는 처리에 대해 부분적으로 고속화를 하고 시스템을 구성하는 각 소프트웨어 모듈에 대한 다중 쓰레드 기반 병렬 처리로 구현함으로써 실시간 처리가 가능하다는 장점도 있다.

하지만, 얼굴 모델 정합에서 발생하는 오차로 인해 배경 영역의 픽셀이 얼굴 텍스처에 섞이는 경우가 발생할 수 있는 점과, 변형으로 인해 드러나는 가려짐 영역이 커짐으로 인해 인페인팅 품질이 저하될 수 있는 점은 보완이 필요하다. 또한, 얼굴 모델이 열려 있기 때문에, 자연스러운 변형을 위해 턱과 목의 경계, 귀 근처, 이마와 같은, 얼굴 모델의 가장자리 부분은 고정시켜야한다는 제약은 존재한다. 향후 품질 저하 현상과 제약을 보완하고, 처리 속도 측면에서 더

욱 최적화함으로써 고품질의 얼굴 편집 애플리케이션을 구현할 계획이다.

참 고 문 헌 (References)

[1] Virtual Plastic Surgery Simulator, https://www.plastic-surgery-simulator.com (accessed July 30, 2018)

[2] Snow.me, https://snow.me (accessed July 30, 2018)

[3] H.-V. Chung and I.-K. Lee, "Image-Based Deformation of Objects in Real Scenes," *Proceedings of International Symposium on Visual Computing*, pp.159－166, 2005, https://doi.org/10.1007/11595755_20.

[4] J. Park, B.-K. Seo and J.-I. Park, "[Poster] Interactive Deformation of Real Objects," *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp.295－296, 2014, https://doi.org/10.1109/ISMAR.2014.6948457.

[5] Y. Zheng, X. Chen, M.-M. Cheng, K. Zhou, S.-M. Hu, and N. J. Mitra, "Interactive Images: Cuboid Proxies for Smart Image Manipulation," *ACM Transactions on Graphics*, Vol.31, No.4, pp.99:1－99:11, Jul. 2012, https://doi.org/10.1145/2185520.2185595.

[6] T. Chen, Z. Zhu, A. Shamir, S.-M. Hu, and D. Cohen-Or, "3-Sweep: Extracting Editable Objects from a Single Photo," *ACM Transactions on Graphics,* Vol.32, No.6, pp.195:1－195:10, Nov. 2013, https://doi.org/10.1145/2508363.2508378.

[7] N. Kholgade, T. Simon, A. Efros, and Y. Sheikh, "3D Object Manipulation in a Single Photograph Using Stock 3D Models," *ACM Transactions on Graphics*, Vol.33, No.4, pp.127:1－127:12, Jul. 2014, https://doi.org/10.1145/2601097.2601209.

[8] N. Haouchine, A. Petit, F. Roy, and S. Cotin, "[Poster] Deformed Reality: Proof of Concept and Preliminary Results," *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp.166－167, 2017, https://doi.org/10.1109/ISMAR-Adjunct.2017.56.

[9] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2Face: Real-Time Face Capture and Reenactment of RGB Videos," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2387－2395, 2016, https://doi.org/10.1109/CVPR.2016.262.

[10] K. Dale, K. Sunkavalli, M. K. Johnson, D. Vlasic, W. Matusik, and H. Pfister, "Video Face Replacement," *ACM Transactions on Graphics*, Vol.30, No.6, pp.130:1－130:10, Dec. 2011, https://doi.org/10.1145/2070781.2024164.

[11] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou, "FaceWarehouse: A 3D Facial Expression Database for Visual Computing," *IEEE Transactions on Visualization and Computer Graphics*, Vol.20, No.3, pp.413－425, Mar. 2014, https://doi.org/10.1109/TVCG.2013.249.

[12] C. Cao, H. Wu, Y. Weng, T. Shao, and K. Zhou, "Real-Time Facial Animation with Image-Based Dynamic Avatars," *ACM Transactions on Graphics*, Vol.35, No.4, pp.126:1－126:12, Jul. 2016, https://doi.org/10.1145/2897824.2925873.

[13] K. Nagano, J. Seo, J. Xing, L.Wei, Z. Li, S. Saito, A. Agarwal, J. Fursund, and H. Li, "paGAN: Real-Time Avatars Using Dynamic Textures," *ACM Transactions on Graphics*, Vol.37, No.6, pp.258:1－258:12, Dec. 2018, https://doi.org/10.1145/3272127.3275075.

[14] J. Park and J.-I. Park, "A Framework for Virtual 3D Manipulation of Face in Video," *Proceedings of the IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp.649－650, 2018, https://doi.org/10.1109/VR.2018.8446445.

[15] S. Ren, X. Cao, Y. Wei, and J. Sun, "Face Alignment at 3000 FPS via Regressing Local Binary Features," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.1685－1692, 2014, https://doi.org/10.1109/CVPR.2014.218.

[16] V. Kazemi and J. Sullivan, "One Millisecond Face Alignment with an Ensemble of Regression Trees," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.1867－1874, 2014, https://doi.org/10.1109/CVPR.2014.241.

[17] M. Kowalski, J. Naruniec, and T. Trzcinski, "Deep Alignment Network: A Convolutional Neural Network for Robust Face Alignment," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2017.

[18] P. Huber, G. Hu, R. Tena, P. Mortazavian, P. Koppen, W. J. Christmas, M. Ratsch, and J. Kittler, "A Multiresolution 3D Morphable Face Model and Fitting Framework," *Proceedings of the 11th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2016.

[19] O. Sorkine and M. Alexa, "As-Rigid-As-Possible Surface Modeling," *Proceedings of the 5th Eurographics Symposium on Geometry Processing*, pp.109－116, 2007.

[20] A. Telea, "An image inpainting technique based on the fast marching method," *Journal of Graphics Tools*, Vol.9, No.1, pp.23－34, 2004, https://doi.org/10.1080/10867651.2004.10487596.

저 자 소 개

박 정 식

- 2010년 : 한양대학교 전자전기컴퓨터공학부 학사
- 2012년 : 한양대학교 전자컴퓨터통신공학과 석사
- 2012년 ~ 현재 : 한양대학교 컴퓨터소프트웨어학과 박사과정
- ORCID : https://orcid.org/0000-0001-9543-4335
- 주관심분야 : 3차원 영상처리, 증강현실, GPGPU

서 병 국

- 2006년 : 한양대학교 전자전기컴퓨터공학부 학사
- 2008년 : 한양대학교 전자컴퓨터통신공학과 석사
- 2014년 : 한양대학교 전자컴퓨터통신공학과 박사
- 2014년 ~ 2016년 : 독일 프라운호퍼 IGD 연구소 박사후 연구원
- 2016년 ~ 현재 : 한국전자통신연구원 선임연구원
- ORCID : https://orcid.org/0000-0002-7257-4615
- 주관심분야 : 3차원 컴퓨터 비전, 증강현실, 인간컴퓨터상호작용

박 종 일

- 1987년 : 서울대학교 전자공학과 학사
- 1989년 : 서울대학교 전자공학과 석사
- 1995년 : 서울대학교 전자공학과 박사
- 1992년 ~ 1994년 : 일본 NHK 방송기술연구소 객원연구원
- 1995년 ~ 1996년 : 한국방송개발원 선임연구원
- 1996년 ~ 1999년 : 일본 ATR 지능영상통신연구소 연구원
- 1999년 ~ 현재 : 한양대학교 컴퓨터소프트웨어학부 교수
- ORCID : https://orcid.org/0000-0003-1000-4067
- 주관심분야 : 증강현실, 계산사진학, 3차원 컴퓨터비전, 인간컴퓨터상호작용