



국민대학교  
전자정보통신대학  
컴퓨터공학부


# 캡스톤 디자인 I

## 종합설계 프로젝트

프로젝트 명	천의얼굴
팀 명	얼굴코디
문서 제목	중간보고서

Version	1.4
Date	2019-04-17

팀원	조승현 (조장)
	김가연
	김상열
	이대현
	이진구
	강남삼
지도교수	김준호 교수

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	천의얼굴	
	<b>팀 명</b>	얼굴코디	
	Confidential Restricted	Version 1.4	2019-APR-17

### CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인 I 수강 학생 중 프로젝트 “천의얼굴”을 수행하는 팀 “얼굴코디”의 팀원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 “얼굴코디”의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

## 문서 정보 / 수정 내역

<b>Filename</b>	중간보고서-천의얼굴.doc
<b>원안작성자</b>	조승현, 김상열, 김가연, 이대현, 이진구
<b>수정작업자</b>	조승현, 김상열, 김가연, 이대현, 이진구, 강남삼

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2019-04-14	이대현	1.0	최초 작성	중간보고서 초안 작성
2019-04-15	조승현	1.1	내용 수정	연구내용 추가
2019-04-16	이진구	1.2	내용 수정	수정된 연구내용 추가
2019-04-17	김가연	1.3	내용 수정	중간결과 및 연구내용 수정
2019-04-18	김상열	1.4	내용 수정	중간보고서 마무리

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	천의얼굴	
	<b>팀 명</b>	얼굴코디	
	Confidential Restricted	Version 1.4	2019-APR-17

## 목 차


1	프로젝트 목표	4
2	수행 내용 및 중간결과	5
2.1	계획서 상의 연구내용	5
2.2	수행내용	7
2.2.1	Face Detection and Head Pose	7
2.2.2	Light Estimation	7
2.2.3	Color Correction	11
3	수정된 연구내용 및 추진 방향	13
3.1	수정사항	13
3.1.1	시나리오	13
3.1.2	기존 시나리오와 차이점	14
3.2	진행상황	15
3.2.1	ARCORE	15
3.2.2	Texture Map	16
3.2.3	Face Recognition	17
3.2.4	감정분석	18
4	향후 추진계획	19
4.1	향후 계획의 세부 내용	19
1.	APPLICATION	19
2.	SERVER	19
4.2	개발 일정	20
5	고충 및 건의사항	20

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	천의얼굴	
	<b>팀 명</b>	얼굴코디	
	Confidential Restricted	Version 1.4	2019-APR-17

## 1 프로젝트 목표

본 프로젝트는 사람의 얼굴 방향, 표정을 기반으로 얼굴을 합성하여 모두가 잘 나온 단체 사진을 얻는 것을 목표로 한다. 수정할 사진 2 장을 선택하는 과정을 자동화하여 사용자의 서비스 이용에 편리함을 제공한다. 사용자는 타 프로그램과 비교하면 3 초간의 동영상을 찍는 비교적 간단한 시나리오로 최종 합성 사진을 얻을 수 있다.

또한, 증강현실 플랫폼인 ARCore 를 적용하여 단순히 2D 이미지를 합성하지 않고 영상으로부터 3 차원 얼굴 모델을 생성한 후 얼굴 표정을 합성한다. 이 방식은 정교한 자동 합성 시스템을 제공하며 증강현실에서 사용되는 ARCore 를 이용한 새로운 합성 시스템 개발을 목표로 한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	천의얼굴	
	팀 명	얼굴코디	
	Confidential Restricted	Version 1.4	2019-APR-17

## 2 수행 내용 및 중간결과

### 2.1 계획서 상의 연구내용

사용자가 보정할 사진(=타겟 이미지) 1 장과 원하는 표정 사진(=소스 이미지) 여러 장을 선택하면 타겟 이미지와 소스 이미지들의 합성 적합도를 계산하여 사용자에게 제공한다. 사용자가 합성 적합도에 따라 소스 이미지를 선택하면 색과 빛을 보정하여 자연스러운 합성 사진을 얻게 하는 것을 목표로 하였다.

이에 따른 세부 내용은 다음과 같다.

1) 타겟 이미지에서 얼굴을 검출하고 선택된 얼굴의 크기와 방향을 계산한다.

OpenCV API 를 활용하여 얼굴 검출 및 Face Landmark 를 계산해 낼 수 있다. Face Landmark 란 얼굴의 특징점을 검출하는 API 로 눈, 코, 입, 턱 부분을 2 차원 상의 좌표로 나타낸다. 눈과 코의 좌표를 통해서 얼굴의 방향을 계산해낼 수 있다.



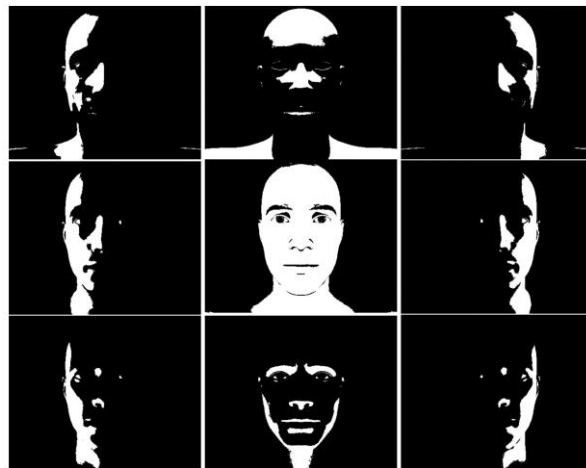
[ 그림 1 ] OpenCV 를 이용한 Face Landmark <sup>1</sup>

<sup>1</sup> [http://www.nada.kth.se/~sullivan/Papers/Kazemi\\_cvpr14.pdf](http://www.nada.kth.se/~sullivan/Papers/Kazemi_cvpr14.pdf)

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	천의얼굴	
	<b>팀 명</b>	얼굴코디	
	Confidential Restricted	Version 1.4	2019-APR-17

2) 타겟 이미지와 소스 이미지의 빛, 색감, 얼굴 방향 등의 정보로 Rule-Based 알고리즘을 통해 합성 적합도를 계산하여 제공한다.

RGB 영역의 이미지를 YUV 영역으로 변환하여 이미지의 명도 값을 분리해 낼 수 있다. 9 개의 방향(상, 하, 좌, 우, 정면, 4 개의 대각방향)에서 얼굴에 빛을 렌더링한 이미지를 활용한다. 이미지 픽셀 값의 유사도 비교를 통해 빛의 방향을 대략적으로 계산해 낼 수 있다. 빛의 유사도, 얼굴 방향의 유사도를 계산하여 합성적합도를 계산한다.



[ 그림 2 ] 빛 방향에 대한 Sample Image

3) 합성 적합도를 통해 선택된 소스 이미지와 타겟 이미지에서 빛을 제거한 후 얼굴 부분을 합성한다. 이후, 빛을 다시 적용하고 색을 보정하여 합성된 이미지를 도출한다.

Homomorphic Filter 를 사용하여 이미지에서 밝기 값을 없앨 수 있다. 이미지를 합성한 후, 2 번에서 분석한 빛을 토대로 새롭게 이미지를 보정할 수 있다. 이미지 합성 후에 OpenCV 에서 제공하는 SeamlessClone 을 사용해 타겟 이미지의 색상과 동일하게 보정된 최종 합성 이미지를 도출한다.

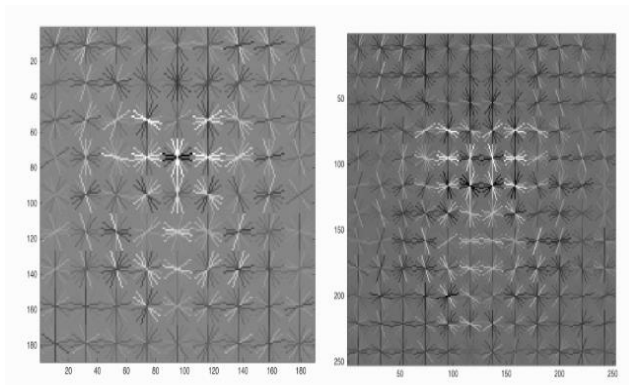
 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	천의얼굴	
	팀 명	얼굴코디	
	Confidential Restricted	Version 1.4	2019-APR-17

## 2.2 수행내용

### 2.2.1 Face Detection and Head Pose

OpenCV API 를 활용하여 Face Detection 을 진행한다. Feature Descriptor 는 HOG 를 사용하며 SVM 분류기를 사용하였다.

Head Pose 또한 OpenCV 를 기반으로 진행된다. 양 눈 끝점, 코끝, 입술의 양쪽 끝 그리고 턱의 Landmark 를 이용하여 Euler Angle 로 얼굴 방향을 계산한다. 얼굴 방향은 YAW, ROLL, PITCH 로 표현할 수 있다



[ 그림 3 ] Hog Descriptor <sup>2</sup>



[ 그림 4 ] Head Pose

### 2.2.2 Light Estimation

1) 이미지의 RGB 영역을 YUV 영역으로 변환하여 기존에 9 가지의 방향에서 찍은 사진과 픽셀 비교를 통해서 빛의 방향을 대략적으로 추정한다. 빛 방향을 추정하기 위해 9 개의 빛 방향 사진에 DCT(Discrete Cosine Transform) 방식을 적용하였다. DCT 는 영상을 공간 영역으로부터 주파수 영역으로 변환하여 우수한 화질로 압축하는 데 사용되는

<sup>2</sup> <http://sklin93.github.io/hog.html>

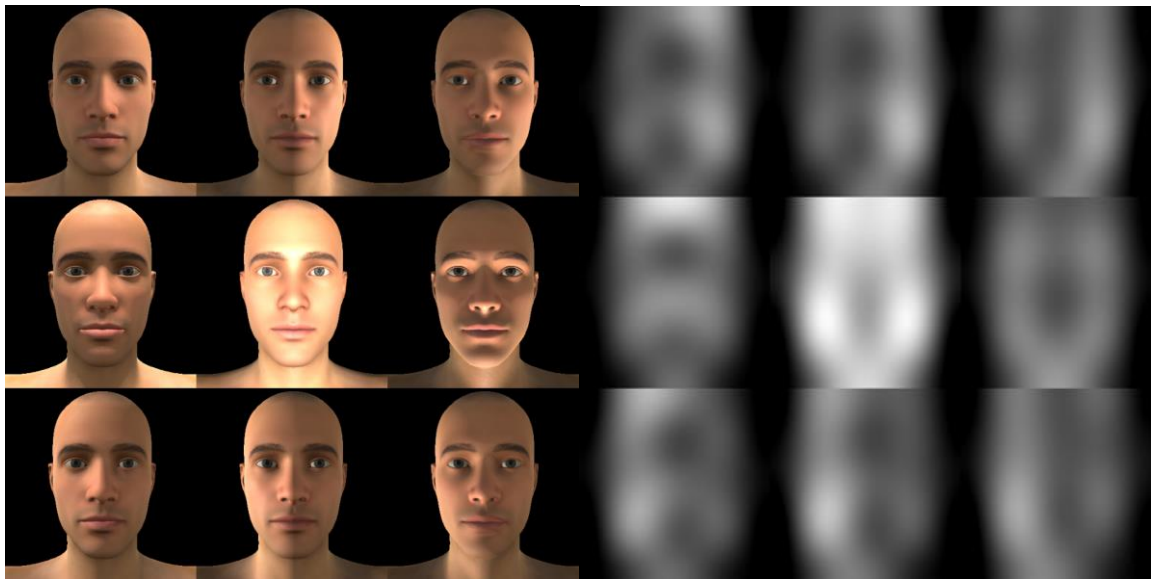
 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	천의얼굴	
	팀 명	얼굴코디	
	Confidential Restricted	Version 1.4	2019-APR-17

알고리즘이다. DCT 로 압축을 진행할 때 coefficient 값을 변환하면 빛에 대한 정보를 추출할 수 있다.

$$C(u,v) = a(u)a(v) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \times \cos\left[\frac{(2x+1)u\pi}{2M}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

[ 식 1 ] N\*N 행렬에서 DCT 공식 <sup>3</sup>

아래의 그림은 DCT 공식에서 10 개의 coefficient 값만 남겨두도록 변형하여 9 개의 방향에서 얼굴에 빛을 비추었을 때 빛 방향을 추출한 결과이다.



[ 그림 5 ] 9 개의 방향에서 빛을 비추는 Image

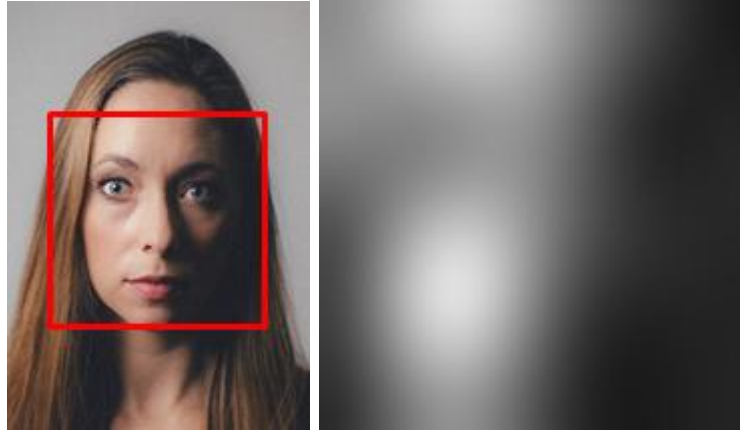
[ 그림 6 ] DCT 를 변환하여 얻은 빛 정보

<sup>3</sup> <https://ieeexplore.ieee.org/document/6936172>



 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	천의얼굴	
	<b>팀 명</b>	얼굴코디	
	Confidential Restricted	Version 1.4	2019-APR-17

DCT 를 같은 방식으로 변형하여 타겟 이미지에서 빛 정보를 추출하였다.



[ 그림 7 ] 좌. 타겟 이미지 (원본 이미지)

[ 그림 8 ] 우. 타겟 이미지로부터 추출한 빛 정보

이후, 빛의 방향을 추출한 9 개의 이미지와 타겟 이미지의 빛 방향을 픽셀별로 비교한다.

빛의 유사도를 측정하기 위해 아래의 수식을 이용하였다.

$$I_p = \frac{\sum_{i=1}^m \sum_{j=1}^n L_p(i, j) \times L_r(i, j)}{\|L_p\| \times \|L_r\|}$$

[ 식 2 ] 두 이미지 사이의 유사도 비교 <sup>4</sup>

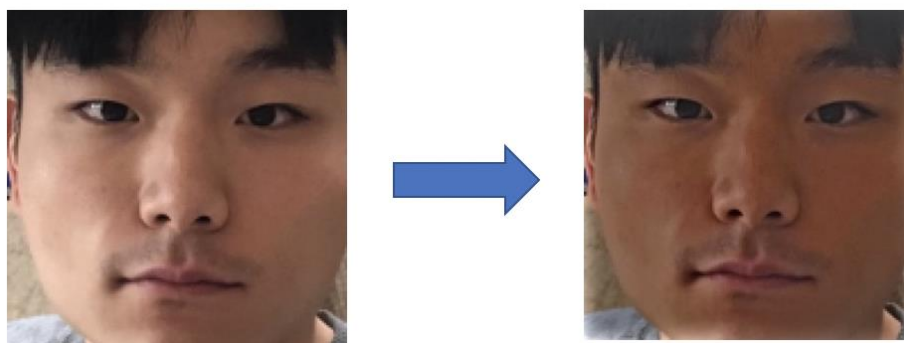
<sup>4</sup> <https://ieeexplore.ieee.org/document/6936172>

유사도 비교를 진행한 결과는 다음과 같다. Index 0 ~ 8 은 위에서 제시한 9 개의 빛 방향을 의미한다.


	similarity	
0	0.002624	[ 그림 6 ]은 9 개의 빛 방향과 타겟이미지의 빛 유사도 비교 결과를 의미한다. 유사도 비교 결과 7 번 빛과 가장 높은 유사도를 가진다. 하지만 두 이미지의 픽셀 단위의 유사도 비교는 같은 얼굴 부위끼리 비교하지 못할 가능성이 높아 공정한 유사도 비교가 이루어지지 않는다.
1	0.002784	
2	0.003024	
3	0.003968	
4	0.004288	
5	0.003808	
6	0.004336	
7	0.006528	
8	0.004400	

[ 그림 9 ] 빛 유사도

2) 원본 이미지에서 Homomorphic Filter 활용하여 밝기를 제거한다. Homomorphic Filter 란 이미지를 주파수 영역으로 분리하여 이미지의 밝기를 보정할 때 사용하는 것으로 아래 그림은 주파수 영역에서의 log 연산으로 이미지의 조명을 제거한 결과이다.



[ 그림 10 ] Light Reduction

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	천의얼굴	
	<b>팀 명</b>	얼굴코디	
	Confidential Restricted	Version 1.4	2019-APR-17

3) 1 번 과정에서 얻은 이진화 영상을 토대로 덧셈 연산을 통해 빛이 보정된 이미지를 얻을 수 있다.



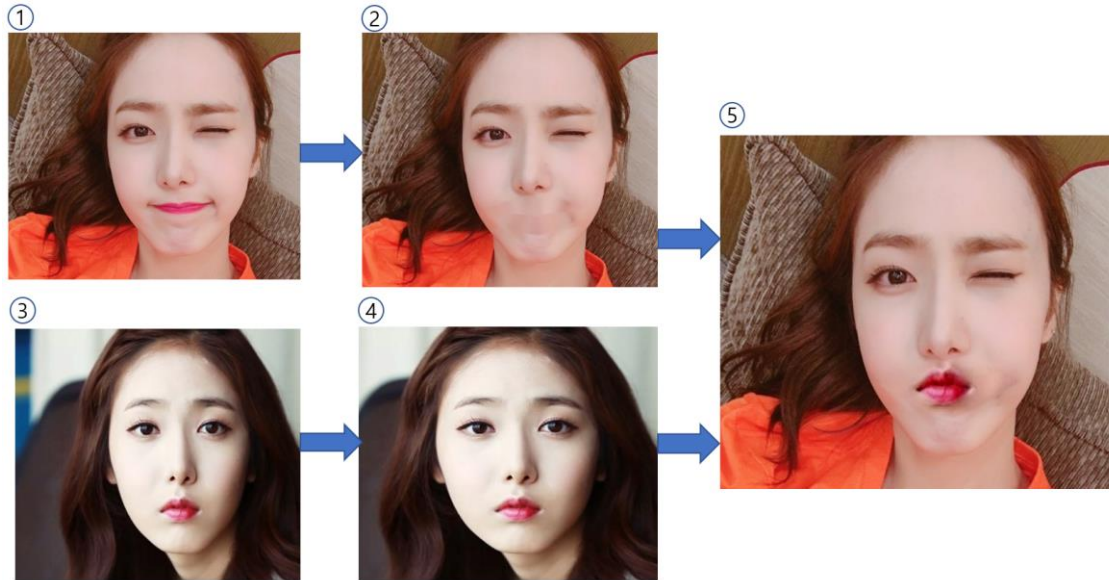
[ 그림 11 ] Light Applying

위 이미지는 동일한 이미지에서 빛을 제거한 후 다른 방향의 빛을 적용하여 생성한 이미지이다. 빛의 유사도 비교를 수행하였지만, 일반 이미지에서는 얼굴의 크기, 위치, 각도가 달라 빛을 픽셀 별로 비교하게 되면 공정한 비교가 이루어지지 않는다. 따라서 부자연스러운 결과가 나오게 된다.

### 2.2.3 Color Correction

Color Correction 은 mask 를 합성한 후, 합성된 부분을 자연스럽게 하기 위하여 색을 보정해주는 과정이다. 아래의 사진은 Color Correction 의 기능을 확인하기 위해 입 부분에 대하여 합성을 진행한 결과이다.

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	천의얼굴	
	<b>팀 명</b>	얼굴코디	
	Confidential Restricted	Version 1.4	2019-APR-17




[ 그림 12 ] Color Correction 진행결과

진행 과정은 다음과 같다.

- 1) 사진 1 을 OpenCV 의 inpaint 함수를 사용하여 입 부분을 입 주변의 색으로 채워 넣어 사진 2 로 만든다.
- 2) 사진 3 의 입을 합성할 사진 1 과 같은 위치에 올 수 있게 transform 을 해주어 사진 4 로 만든다.
- 3) 사진 2 와 사진 4 의 입 부분만 OpenCV 의 SeamlessClone 을 사용하여 사진 5 를 만든다.

결과물[사진 5]을 통해 다른 환경의 두 사진의 색을 보정하는 것은 한계가 있다는 것을 확인하였다.

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	천의얼굴	
	<b>팀 명</b>	얼굴코디	
	Confidential Restricted	Version 1.4	2019-APR-17

### 3 수정된 연구내용 및 추진 방향

#### 3.2 수정사항

##### 3.1.1 시나리오

위 2.2 수행내용에 제시된 공통된 문제점은 공정하지 못한 비교이다. 공정하지 못한 비교란 픽셀 단위에서 서로 다른 얼굴 부분을 대등하게 비교가 불가능하다는 것이다. 이러한 이유로 얼굴에 대한 3D Mesh 및 Texture Image 를 생성하여 문제를 해결하고자 한다.

또한, 전혀 다른 주변 환경에서 찍힌 2 장의 사진일 경우, 주변 환경에 맞도록 사진을 보정하는 것만으로도 합성 결과가 좋지 못했다. 따라서 동영상으로부터 나뉜 프레임으로 사진을 합성하는 것으로 시나리오를 수정하였다. 주변 환경을 비슷하게 하여 최대한 이미지의 보정 요소를 제한하려 한다.

이에 따라 제안서의 다음과 같이 시나리오를 수정하였다.

- 1) 3 초간의 비디오 촬영을 통해 30 장의 Still Image 와 얼굴의 3D Mesh 정보와 Texture Image 를 획득한다.
- 2) 30 장의 사진 중 가장 많은 사람이 정면으로 나온 사진을 Base Image 로 선정한다. 만일, Base Image 로 선정된 사진이 사용자의 마음에 들지 않을 경우 직접 선택할 수 있다.
- 3) Base Image 를 제외한 29 장의 사진에 대해 사람별로 같은 Directory 에 저장한다. 각 사람에 대하여 감정분석을 수행하여 행복 수치가 가장 높은 (웃고 있는) 얼굴을 Source Image 로 선정한다.
- 4) Texture Image 를 활용하여 Source Image 와 Base Image 의 빛 유사도를 비교한다. Base Image 의 빛을 Source Image 에 적용하여 빛을 보정한다. 같은

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	천의얼굴	
	<b>팀 명</b>	얼굴코디	
	Confidential Restricted	Version 1.4	2019-APR-17

방식으로 색에 대한 보정도 Texture Image 를 활용하여 진행한다.

- 5) 빛과 색에 대하여 보정이 완료된 Source Texture Image 를 활용하여 3 차원으로 모델링 후에 Mask Image 를 Base Image 에 합성한다.

### 3.1.2 기존 시나리오와 차이점

기존 시나리오상에서 사용자가 원하는 표정의 소스 이미지를 선택하는 번거로움이 있었다면, 사용자는 3 초간 동영상을 촬영하여 30 장의 프레임을 생성한다. 30 장의 프레임과 감정인식 인공지능을 활용하여 모두가 행복한 표정을 자동으로 합성해주는 유저 시나리오로 변경하였다. 이는 기존 시나리오보다 간단한 유저 시나리오이며, 비슷한 주변 환경들로 합성한다는 점이 장점이다.

또한, 2D 상에서의 단순한 이미지 합성으로 자연스럽지 못한 결과물이 나왔던 기존 시나리오와 달리, 수정된 시나리오에서는 ARCore 를 활용해 얻어낸 3D Mesh 정보와 Texture Map 이용하여 3D Model 을 만든다. Texture Map 을 이용하여 빛과 색을 보정하면 더 공정한 픽셀 비교를 할 수 있기 때문에 만족스러운 결과물을 얻을 것으로 예상된다. 합성 단계는 단순히 얼굴을 덮어씌우는 과정이 아닌, 보정이 끝난 Texture Map 을 활용하여 3 차원으로 모델링 후에 Mask Image 를 Base Image 에 합성한다.

수정된 시나리오는 동일한 환경에서 촬영된 사진을 합성하기 때문에 기존 시나리오에서 문제가 되었던 빛, 해상도가 해결되었고 ARCORE 를 통해 Unwrapped Image 를 기반으로 합성하여 보다 정교한 합성이 가능해질 것이다.

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	천의얼굴	
	<b>팀 명</b>	얼굴코디	
	Confidential Restricted	Version 1.4	2019-APR-17

## 3.2 진행상황

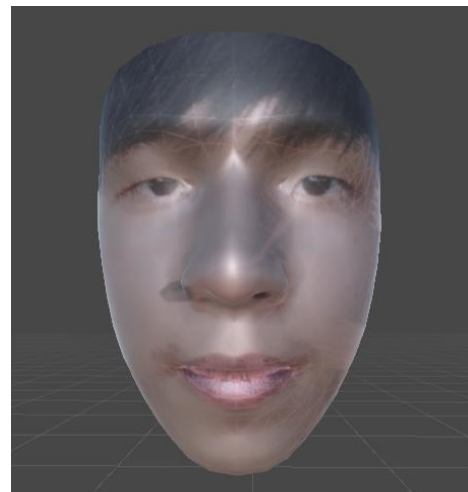
### 3.2.1 ARCORE

현재 구글에서 제공하는 ARCore 최신 버전은 모바일 상에서 작동 가능한 Tensorflow Lite 를 사용하여 영상에서 3 차원 얼굴 Mesh 정보를 제공하고 있다. 3 차원 Mesh 정보를 활용하여 공정한 픽셀 비교가 가능하다.

아래 사진은 ARCore 카메라로 찍은 얼굴을 3 차원으로 복원한 사진이다. ARCore 의 Unwrap 기능을 이용하면 정교한 보정과 합성이 가능하다.



[ 그림 13 ] 3 차원 얼굴

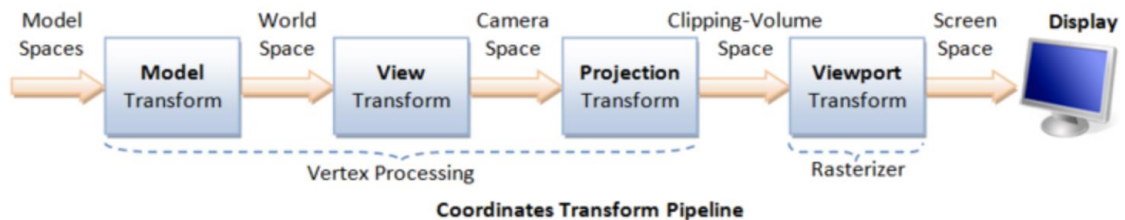


[ 그림 14 ] 3D model

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	천의얼굴	
	팀 명	얼굴코디	
	Confidential Restricted	Version 1.4	2019-APR-17

### 3.2.2 Texture Map

ARCore 를 통해 얻어낸 Mesh 의 3 차원 Vertice 좌표를 활용하여 Polygon 을 구성하는 Triangle 로 관심 영역(ROI)을 얻어낼 수 있다. 관심 영역을 UV 좌표로 변환하여 Texture Image 를 얻어낼 수 있다. 이때 Local Space to Screen Space 와 Affine 변환이 필요하다. ARCore 가 제공하는 좌표들은 Model 자체의 Local Space 로 이 좌표를 Screen Space 로 변환하여야 한다. Affine 변환은 직선의 길이, 평행성을 보존하는 변환으로 3 차원의 얼굴을 2 차원의 Texture Map 으로 변환하기 때문에 Affine 변환을 사용하여 공간성을 유지한다.



[ 그림 15 ] Model 의 좌표계 <sup>5</sup>



[ 그림 16 ] Unwrapped Image

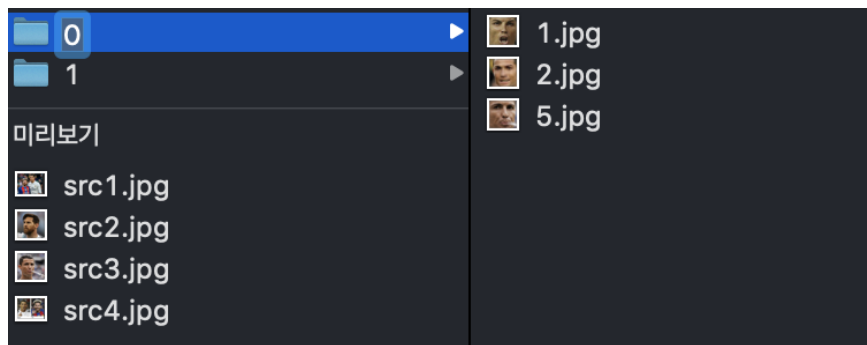
<sup>5</sup> [https://www.ntu.edu.sg/home/ehchua/programming/opengl/cg\\_basicstheory.html](https://www.ntu.edu.sg/home/ehchua/programming/opengl/cg_basicstheory.html)



 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	천의얼굴	
	팀 명	얼굴코디	
	Confidential Restricted	Version 1.4	2019-APR-17

### 3.2.3 Face Recognition

Python 에서 제공하는 Face Recognition API 를 사용하였으며 각 얼굴을 인코딩하여 128 차원의 Feature Vector 로 나타내고 각 Vector 사이의 거리가 0.6 이하일 경우 동일 인물로 판단하게 된다. Base Image 의 각 얼굴에 대하여 사진을 저장하고 29 장의 Source Image 에서 동일 인물별로 Directory 를 만들어 얼굴 Image 를 저장한다.



[ 그림 16 ] Face Recognition 결과

### 3.2.4 감정분석

기존의 시나리오에서 사용자가 원하는 얼굴의 사진을 찾아서 합성을 진행하는 과정은 굉장히 복잡했다. 따라서 감정분석 AI 를 활용하여 단체 사진 속 모두의 얼굴을 행복한 얼굴로 합성을 진행한다.

감정분석 API 는 Microsoft 에서 개발한 Face API 를 사용할 예정이다. Face API 는 사진 속에서 사람의 얼굴을 인식한 뒤, 인식된 사람의 표정을 읽어 분노, 경멸, 혐오 공포, 행복, 무표정, 슬픔, 놀람의 표정의 정도를 0 에서 1 사이의 값으로 나타낸다. 이 중 행복(Happiness) 값이 가장 높은 값으로 합성이 진행된다. 다음은 Face API 의 결과물이다.




	0	1	2	3	4
anger	0.0	0.0	0.0	0.0	0.0
contempt	0.014	0.0	0.0	0.001	0.0
disgust	0.0	0.0	0.0	0.0	0.0
fear	0.0	0.0	0.0	0.0	0.0
happiness	0.005	0.921	0.016	0.002	0.039
neutral	0.954	0.058	0.984	0.988	0.96
sadness	0.003	0.0	0.0	0.009	0.0
surprise	0.023	0.02	0.0	0.0	0.0

[ 그림 17 ] 감정분석에 사용한 이미지 <sup>6</sup>

[ 그림 18 ] 감정분석 결과

<sup>6</sup> <https://www.facebook.com/IRENEpageS2/>

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	천의얼굴	
	<b>팀 명</b>	얼굴코디	
	Confidential Restricted	Version 1.4	2019-APR-17

## 4 향후 추진계획

### 4.2 향후 계획의 세부 내용

#### 1. APPLICATION

- 1) 사람들이 동영상 촬영 버튼을 누르고 3 초 동안 30 장의 Still Image 와 각 인물에 대한 Unwrap Image 를 서버로 전송한다.
- 2) 서버에서 결정된 Base Image 를 보여주고 사용자가 마음에 들지 않을 시 변경이 가능하도록 한다.
- 3) 모든 기능을 구현하고 최종 단계로 App UI 를 구성하여 완성된 애플리케이션을 제공한다.

#### 2. SERVER

- 1) Base Image 는 30 장의 Still Image 중 정면을 보고 있는 사람이 가장 많은 이미지를 기본값으로 설정한다. Normal Vector 를 가지고 있는 3 차원의 Vertex 중 코끝의 Normal Vector 를 이용하여 얼굴의 방향을 계산한다.
- 2) 30 장의 Still Image 에서 인물마다 사진을 Grouping 한 후, 각 얼굴에 대해 행복 지수가 가장 높은 얼굴을 계산하여 각 인물의 Source Image 를 만든다. 만약 5 명의 단체 사진이라면 30 장의 Still Image 에서 5 장의 Source Image 가 생성된다.
- 3) Source Image 의 Texture Image 를 Base Image 의 환경에 맞도록 보정한다. 보정 요소에는 얼굴각도, 빛, 색이 포함된다. Texture Image 상에서 보정을 진행하며 공정한 픽셀 비교를 통해 빛과 색에 대한 보정이 이루어진다.
- 4) 보정된 Texture Image 를 Base Image 에 적용하여 합성한다.

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	천의얼굴	
	<b>팀 명</b>	얼굴코디	
	Confidential Restricted	Version 1.4	2019-APR-17

## 4.2 개발 일정

날짜	수행 내용
04.21 ~ 04.27	3D modeling 구현, 서버 자동화
04.28 ~ 05.04	얼굴 합성 구현, 서버 자동화
05.05 ~ 05.11	App UI & 얼굴 합성 구현
05.12 ~ 05.18	만족도 조사 시행
05.19 ~ 05.24	부족한 기술(버그) 수정
05.26 ~ 05.31	최종 시연 정리

## 5 고충 및 건의사항

- 1) 사용하려는 ARCore 의 Augmented Face API 가 올해 초에 나온 ARCore 1.7 버전이므로 관련된 정보가 부족했다.
- 2) 합성 기능을 구현하다 보니 결과가 생각보다 좋지 않아 시나리오 재구성에 많은 시행착오를 겪었다.
- 3) ARCore 를 지원하는 기기는 안드로이드인데, 팀원 대부분이 IOS 기기를 사용하고 있었기 때문에 안드로이드 기기를 구하는데 많은 시간이 걸렸다.