

2019년 2월
석사학위논문

Conditional Generative
Adversarial Network 기반
카툰원화의 Line drawing 추출

조선대학교 대학원

컴퓨터공학과

유 경 호

Conditional Generative
Adversarial Network 기반
카툰원화의 Line drawing 추출

Extraction of line drawing from Cartoon painting
based on Conditional Generative Adversarial Network

2019년 2월 25일

조선대학교 대학원

컴퓨터공학과

유 경 호

Conditional Generative
Adversarial Network 기반
카툰원화의 Line drawing 추출

지도교수 양 희 덕

이 논문을 공학석사학위신청 논문으로 제출함




2018년 10월

조선대학교 대학원

컴퓨터공학과

유 경 호

유경호의 석사학위논문을 인준함

위원장	조선대학교 교수	<u>강문수</u>	
위원	조선대학교 교수	<u>최우열</u>	
위원	조선대학교 교수	<u>양희덕</u>	

2018년 11 월

조선대학교 대학원

목 차

ABSTRACT

I . 서 론	1
A. 연구의 배경 및 목적	1
B. 연구 내용	4
II . 관련연구	5
A. 기하학적 특징선	5
B. Autoencoder	11
C. Convolutional Neural Network	13
D. Generative Adversarial Network	15
E. 드로잉 기반 모델링	17
III . Line drawing 추출, 검출, 생성	19
A. 3D 모델의 line drawing 추출	19
B. Line drawing 검출 및 생성	24
IV . 실험 및 결과 분석	26
A. 실험 환경	26
B. 실험 결과 분석	28
V . 결론	34
참고문헌	35

표 목 차

표 1. 실험 환경 및 설계	26
표 2. Subdivision surface 수행 횟수에 따른 결과	29
표 3. Conditional GAN을 이용한 line drawing 추출 결과 - 1	30
표 4. Conditional GAN을 이용한 line drawing 추출 결과 - 2	31
표 5. Conditional GAN을 이용한 line drawing 추출 결과 - 3	32

그림 목 차

그림 1. 2D 이미지와 3D 모델에서 특징선 추출	2
그림 2. 기하학적 특성을 보이는 카툰 원화에서 특징선 검출	3
그림 3. Line drawing 추출의 전체 과정	4
그림 4. 다양한 특징선의 비교 [8]	5
그림 5. 곡면에서 contour 생성기와 contour [6]	6
그림 6. 곡면에서 contour와 suggestive contour [6]	7
그림 7. Suggestive contour의 정의를 만족하는 점 [6]	8
그림 8. 점 a' 보다 점 b' 에서 더 큰 최대 뷰-종속 곡률 [8]	10
그림 9. Autoencoder의 구조 [11]	11
그림 10. Convolutional Neural Network의 기본 구조	13
그림 11. 컨볼루션 계층의 연산 예	13
그림 12. 최대 풀링의 예	14
그림 13. Generative Adversarial Network의 기본구조	15
그림 14. 딥러닝 기술을 이용한 드로잉 기반 모델링 [15]	17
그림 15. 스케치기반 3D 모델 검색의 아키텍처(상)와 출력 예(하) [16]	18
그림 16. 직교 투영 방식을 사용한 line drawing 이미지 생성	19
그림 17. 특징선의 조합으로 나타낸 line drawing	20
그림 18. 3D 캐릭터 모델의 subdivision surface 전(좌) · 후(우) 비교	21
그림 19. Subdivision surface 횡수에 따른 곡률의 변화	21
그림 20. Subdivision surface 횡수에 따른 3D 모델의 특징선 추출	23
그림 21. 2D 카툰이미지에서 line drawing 생성을 위한 Conditional GAN 학습	24
그림 22. 학습에 사용되는 카툰 이미지(좌)와 line drawing(우)	27
그림 23. 2D 이미지에서 기하학적 속성을 나타내는 line drawing 추출	28
그림 24. Ground Truth와 Output의 PSNR 그래프	33

ABSTRACT

Extraction of line drawing of Cartoon painting based on Fully Convolution Encoder–Decoder Network

Yu, Kyoungho

Advisor : Prof. Yang, HeeDeok, Ph.D.

Department of Computer Engineering

Graduate School of Chosun University

Recently, 3D contents used in various fields have been attracting people's attention due to the development of virtual reality and augmented reality technology. In order to produce 3D contents, it is necessary to model the objects as vertices. However, high-quality modeling is time-consuming and costly. In this work, we propose a method, based on conditional adversarial network, for automatic extraction of line drawings that show the geometrical characteristics of 3D models in 2D cartoon painting.

Drawing-based modeling among non-photorealistic expressions is a technique that shortens the time required for modeling. It refers to creating a 3D model based on a user's line drawing or sketch. Line drawings are used to represent the shape of objects using a line that is a minimum of data. Line drawings require feature line extraction to determine which part of the object should be represented as a line. Extracting feature lines provides more information about a 3D model.

In order to convert a 2D character into a 3D model, it is necessary to express it as line drawings through feature line extraction. The extraction of consistent line drawings from 2D cartoon cartoons is difficult because the styles and techniques differ depending on the designer who produces

them. Therefore, it is necessary to extract the line drawings that show the geometrical characteristics well in 2D cartoon shapes of various styles. This study proposes a method of automatically extracting line drawings. The 2D Cartoon shading image and line drawings are learned by using conditional adversarial network model, which is artificial intelligence technology and outputs 2D cartoon artwork of various styles. Experimental results show the proposed method in this research can be obtained as a result of the line drawings representing the geometric characteristics when a 2D cartoon painting as input.

1. 서론

A. 연구의 배경 및 목적

3D 컴퓨터 그래픽스 기술로 만들어진 콘텐츠는 사용자가 현실에서 경험할 수 없는 것을 체험 할 수 있기 때문에 게임, 영화, 애니메이션, 교육 등 다양한 분야에서 사용되고 있다. 3D 콘텐츠는 2D 콘텐츠와 다르게 사용자가 콘텐츠 구성 요소와 직접 상호작용할 수 있어 몰입감과 현실감을 느낄 수 있다. 최근 가상현실(Virtual reality)과 증강현실 기술(Augmented reality)의 발전으로 사람들이 손쉽게 3D 콘텐츠를 접할 수 있고 그 결과 3D 콘텐츠에 대한 사람들의 관심이 높아지고 있다.

3D 콘텐츠를 제작하기 위해서는 콘텐츠를 구성하는 요소를 3D 공간에 정점(Vertex)으로 나타내는 3D 모델링이 필수이며, 고품질의 3D 모델링은 시간과 비용이 많이 소요된다. 비사실적 표현 기술 중 드로잉 기반 모델링은 3D 모델링에 소요되는 시간을 단축시키는 기술이다. 이 기술은 line drawing이나 sketch를 기반으로 3D 모델을 생성하는 것을 말한다[1].

Line drawing이란 사물의 형태를 선(Line)을 이용하여 나타내는 것을 말한다[2]. 비사실적 렌더링(Non-photo realistic rendering, NPR)은 현실의 사물을 사실적으로 표현하는 전통적인 컴퓨터 그래픽스와 달리 현실의 사물을 사실적으로 표현하지 않는 분야이다. 비사실적 렌더링에서 line drawing은 가장 기본적인 요소로서 어떤 부분을 선으로 나타낼 것인지 여부를 결정하는 특징선 추출이 필요하다. 특징선 추출방법으로는 2D 영상에서 특징선 추출하는 방법과 3D 모델에서 특징선을 추출하는 방법이 있다. 3D 모델에서 특징선을 추출하는 방법은 2D 영상을 사용할 때보다 3D 모델에 대한 정보를 더 많이 추출할 수 있고 라인의 두께나 스타일을 유연하게 조절 가능하다[1-3].

카툰 이미지를 제작하는 방법에는 디자이너가 직접 그리는 방법과 컴퓨터 그래픽스 기술을 사용하는 방법이 있다. 디자이너가 제작하는 방법은 세부적인 부분(재질, 음영 등)을 직접 그려 카툰 이미지를 제작한다. 반면 컴퓨터 그래

픽스 기술을 사용하는 방법은 3D 모델링으로 형태를 만들고 텍스처 매핑(Texture mapping)을 통해 사물의 질감을 나타낸다. 그 다음 NPR 기법 중 하나인 카툰 렌더링을 통해 카툰 이미지를 제작한다[4,5].

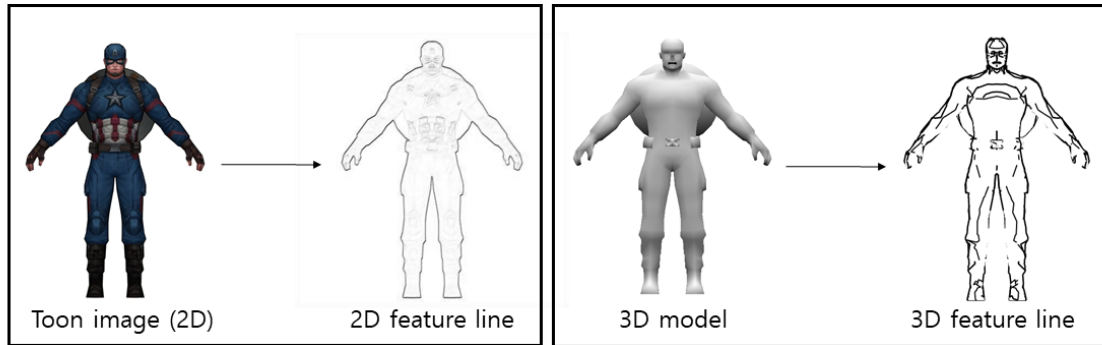


그림 1. 2D 이미지와 3D 모델에서 특징선 추출

2D 카툰 이미지에서 캐니 에지 검출(Canny edge detection)과 같은 2D 이미지의 특징선 추출은 3D 모델의 곡률이나 각 정점과 시점과의 관계에 따라 생성되는 특징선과 나타내는 부분이 다르다. 그림 1에서 볼 수 있듯이 2D 카툰 이미지에서 추출한 특징 선은 3D 모델의 기하학적 속성을 나타내지 않는다.

2D 카툰 원화에서 캐릭터를 3D모델로 변환하기 위해서는 특징선 추출을 통해 line drawing으로 표현하는 것이 필요하다. 2D 카툰 원화에서 3D모델의 기하학적 속성을 갖는 특징 선을 추출하는 것은 어려운 작업이며, 2D 카툰 원화는 제작하는 디자이너에 따라 스타일과 기법이 다르기 때문에 일관성 있는 특징선 추출이 힘들다. 따라서 본 연구는 그림 2와 같이 딥러닝을 사용하여 다양한 스타일의 2D 카툰 원화에서 기하학적 속성을 갖는 line drawing을 추출하는 방법을 제안한다.

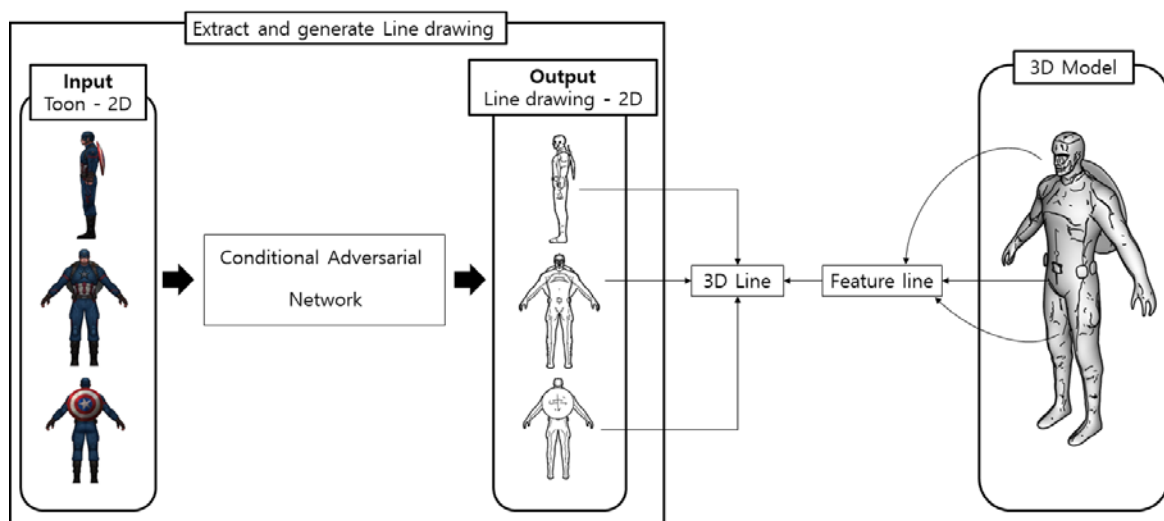


그림 2. 기하학적 특성을 보이는 카툰 원화에서 특징선 검출

B. 연구 내용

본 연구는 Conditional GAN(Generative adversarial network) 모델을 사용하여 카툰원화에서 line drawing을 자동으로 추출하는 방법을 제안한다. Conditional GAN 모델을 2D 카툰 이미지와 line drawing으로 학습시켜 다양한 스타일의 2D 카툰 원화를 입력으로 주었을 때 자동으로 line drawing을 추출한다. 제안하는 2D 카툰 원화에서 line drawing 추출 과정은 그림 3과 같다.

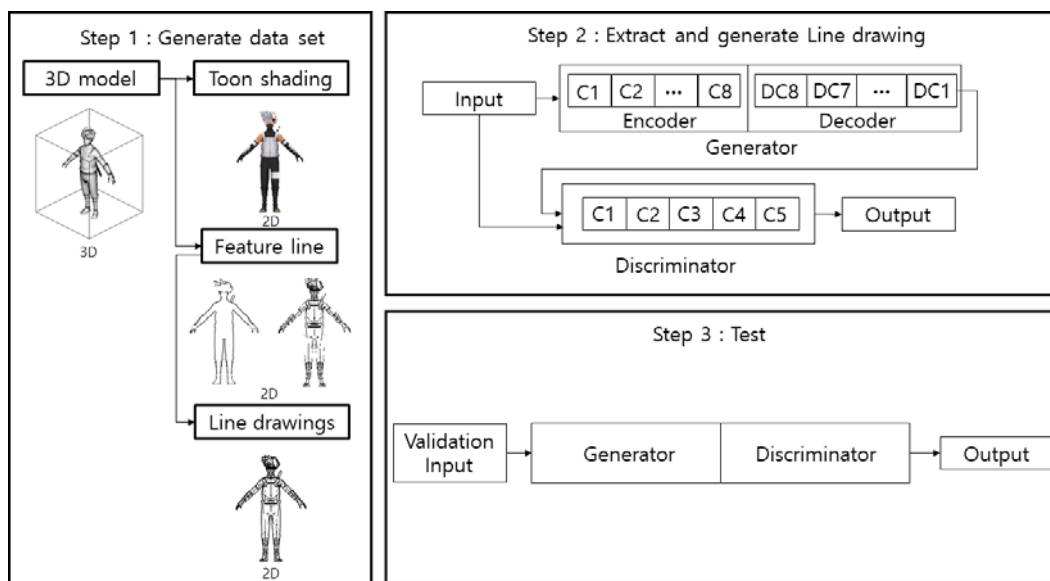


그림 3. Line drawing 추출의 전체 과정

첫 번째 단계에서는 데이터 셋 제작 단계로 3D 모델에서 2D 카툰 이미지와 line drawing 이미지를 생성한다. 카툰 이미지를 생성하기 위해 3D 모델을 만화 처럼 보이도록 NPR 기법 중 하나인 툰 셰이딩을 사용한다. Line drawing 이미지를 생성하기 위해 3D 모델의 기하학적 속성을 나타내는 다양한 특징선을 사용하여 line drawing 이미지를 생성한다. 두 번째 단계에서는 Conditional GAN 모델을 이용하여 데이터 셋을 학습시키는 과정이다. 학습 모델은 Conditional GAN 중 하나인 pix2pix를 수정하여 사용한다. 세 번째 단계에서는 발리데이션 셋을 이용하여 학습 결과를 평가한다.

II. 관련연구

A. 기하학적 특징선

1. 특징선의 종류

특징선(Feature line)은 사물의 어떤 위치에 선을 그릴지 결정하는 것을 말하며 NPR분야에서 기본적인 요소이다. 특징선의 종류는 contours, suggestive contours, ridges, valleys, apparent ridges가 있으며 이와 같은 특징선에 대한 개념이 정립된 이후로 특징선 추출의 효율성과 어떤 형태로 그릴지에 대한 연구가 진행되고 있다[6-8]. 그림 4와 같이 특징선은 기하학적 속성과 3D 모델의 형태에 따라 표현하는 부분이 다르다. 이렇기 때문에 사람의 시각에 따라 어떤 특징선이 미적 묘사와 사실적 묘사를 잘 나타내는지에 대해 평가가 이루어졌다. 그 결과 suggestive contours와 apparent ridges가 대체적으로 형상을 잘 묘사하는 것으로 나타났다[9].

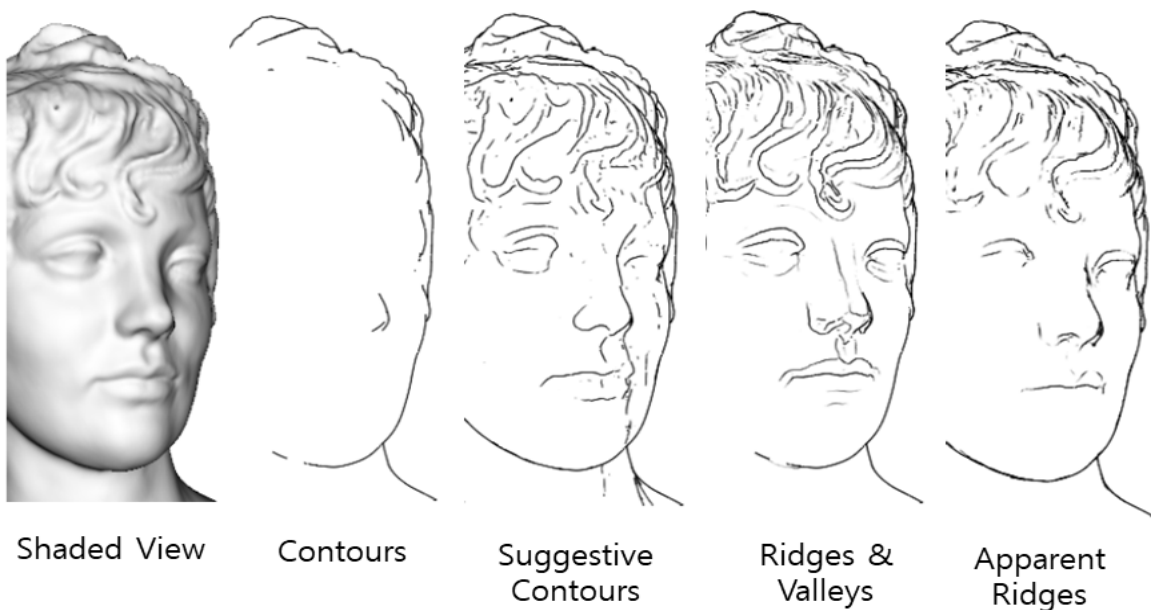


그림 4. 다양한 특징선의 비교 [8]

2. Contours, Suggestive contours

Contours는 뷰-종속적인 선으로 contour 생성기에 의해 생성된 점들 중 2D 이미지에 투영되어 보이는 점들의 집합이다. 그림 5와 같이 곡면의 표면에서 실선과 점선으로 표현된 부분이 contour 생성기에 의해 생성된 점들이다.

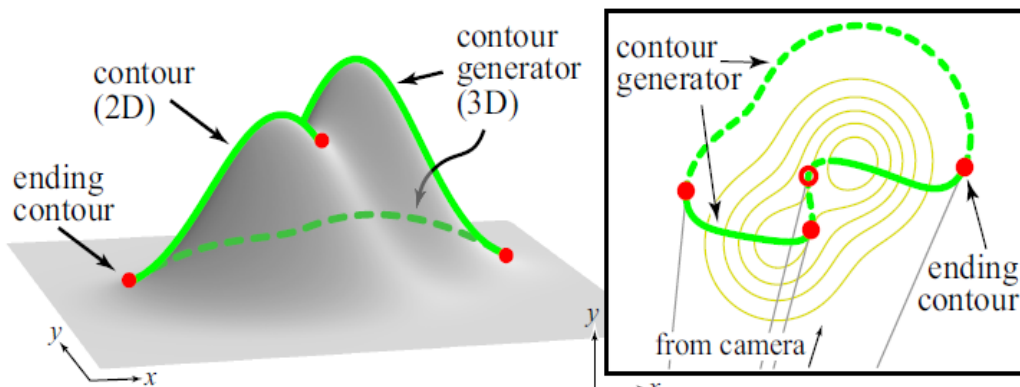


그림 5. 곡면에서 contour 생성기와 contour [6]

생성된 점들 중 카메라 방향에서 보는 2D 이미지에 투영되어 보이는 부분이 contour이다. contour 생성기는 표면을 S 라 하고 S 위의 점 p 가 있을 때 식 (1)을 만족하는 점들의 집합이다. $n(p)$ 는 점 p 의 법선 벡터이고, $v(p)$ 는 점 p 에서 시점 c 로 향하는 뷰 벡터이다[3-5].

$$n(p) \cdot v(p) = 0 \quad (1)$$

그림 4를 보면 contours 만으로는 내부의 세세한 부분까지 나타내지 못하는 단점이 있다. 형상에 세세한 부분을 표현하기 위해 contours와 함께 사용되어 형상의 정보를 효과적으로 제공할 수 있는 특징선으로 suggestive contours가 있다. 그림 4에서 contours와 suggestive contours를 비교하면 suggestive contours는 contours가 표현하지 못하는 부분까지 세세하게 표현할 수 있다[6,7].

Suggestive contours는 뷰-종속적인 선으로 suggestive contour 생성기에 의해 생성된 점들 중 2D 이미지에 투영시켰을 때 이미지에 보이는 점들의 집합이다. suggestive contour는 그림 6의 오른쪽과 같이 파란색 실선과 점선으로 나타낸 점들의 집합이며 이 중 카메라 방향으로 2D 이미지에 투영시켰을 때 contours의 ending contours에서 연장된 선으로 표현된다[6,7].

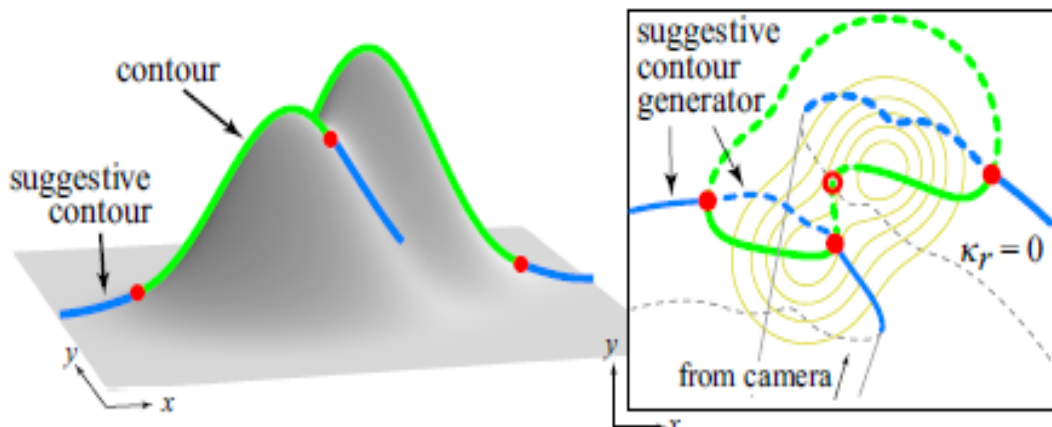


그림 6. 곡면에서 contour와 suggestive contour [6]

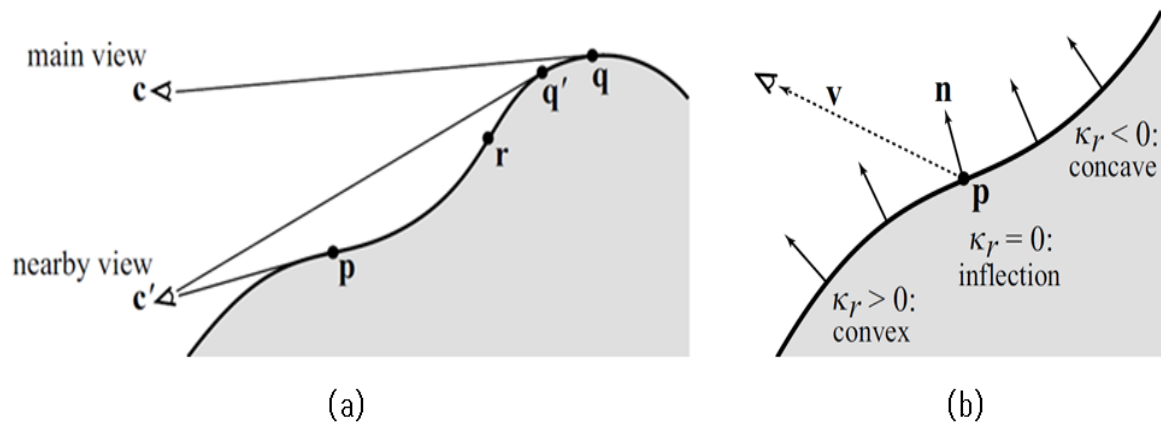


그림 7. Suggestive contour의 정의를 만족하는 점 p [6]

Suggestive contour의 첫 번째 정의는 근처 시점에서 contour이다. 그림 7 (a)에서 시점 c 에서 점 q 는 식 (1)에 의해 contour이다. 시점이 c 에서 c' 으로 이동할 때 contour가 점 q 에서 점 q' 로 미끄러지게 된다. 그러다 시점이 c' 이 되었을 때 점 p 에서 contour가 갑자기 나타나게 된다. 점 p 는 근처 시점 c' 에서 볼 때 contour이지만 시점 c 에 인접한 시점에서는 contour가 아니다. 그래서 주시점인 시점 c 에서 보았을 때 점 q 는 contour이며 점 p 는 suggestive contour이다.

두 번째 정의는 w 방향으로 $n \cdot v$ 의 최솟값을 갖는 점의 집합이다. 방향벡터 w 는 뷰 벡터 v 를 점 p 의 접평면에 투영시킨 벡터이다. w 방향으로 $n \cdot v$ 가 최솟값이 되려면 $n \cdot v \neq 0$ 이어야 하며 이는 법선 벡터 n 과 뷰 벡터 v 가 거의 수직에 가까워야 한다.

세 번째 정의는 반경 곡률(Radial curvature) κ_r 이 0인 지점이며 방향벡터 w 방향으로 반경 곡률 κ_r 의 미분 값인 방향 미분계수 $D_w \kappa_r$ 이 0보다 큰 지점이다. 반경 곡률 κ_r 이 0인 지점은 변곡점(그림 7 (b))이다. 그림 7 (a)에서 변곡점은 점 p 와 r 이다. 하지만 점 r 은 $D_w \kappa_r > 0$ 을 만족하지 않기 때문에 suggestive contour가 아니다[6,7].

3. Ridges, Valleys, Apparent ridges

Ridges와 valleys는 뷰-독립적인 특징선이며 주방향으로 주 곡률이 지역적으로 최댓값인 점들의 집합이다. 점 m 에서 곡률 연산자 S 는 식 (2)으로 정의한다.

$$S(r) = D_r n \quad (2)$$

여기서, n 은 법선 벡터이며 $D_r n$ 은 접평면에서 벡터 r 방향으로 방향미분 계수이다. 표면의 모든 점에 대해서 최대 및 최소 주 곡률 κ_1 과 κ_2 는 $|\kappa_1| > |\kappa_2|$ 일 때 S 의 고윳값이다. ridges와 valleys는 최대 및 최소 주 곡률 방향과 일치하는 벡터 e_1 과 e_2 에서 방향 미분 계수 $D_e \kappa_1 = 0$ 인 점의 집합이다. ridges는 $\kappa_1 > 0$, valleys는 $\kappa_2 < 0$ 인 값을 갖는다.

Apparent ridges는 뷰-종속적인 선이며 뷰-종속 주방향에서 뷰-종속 곡률이 최댓값인 점들의 집합이다. 뷰-종속 곡률은 시점에서 얼마나 굴절되었는지를 나타내며 그림 8과 같이 3차원 공간에 있는 물체를 viewing 스크린 평면에 투영시켜 얻을 수 있다. 물체 M 이 $M \subset R^3$ 이며 viewing 스크린 평면을 V 라 할 때 점 m 은 $m \in M$ 이고, 점 m' 은 점 m 을 viewing 스크린 평면 V 에 투영한 것으로 $m' \in V$ 이다. 점 m' 에서 뷰-종속 곡률 연산자 Q 는 식 (3)로 정의한다[8].

$$Q(s) = D_s n' \quad (3)$$

여기서, $D_s n'$ 는 스크린 평면에서 벡터 s 방향으로 n' 을 미분한 방향 미분계수이다. n' 은 viewing 스크린 평면에 투영된 점 m 의 법선 벡터이다. $Q(s)$ 는 스크린에서 벡터 s 를 따라 이동할 때 법선이 어떻게 변경되는지를 나타내는 접선 평면의 벡터이다. apparent ridges는 최대 뷰-종속 곡률이 최댓값인 점의 집합이므로 최대 뷰-종속 곡률 q_1 은 식 (4)로 정의한다.

$$q_1 = \max_{\|s\|=1} \|Q(s)\| \quad (4)$$

최대 뷰-종속 곡률 q_1 은 Q 의 최댓값이고 벡터 s 가 Q 의 최대 단일 벡터일 때 최댓값을 갖는다.

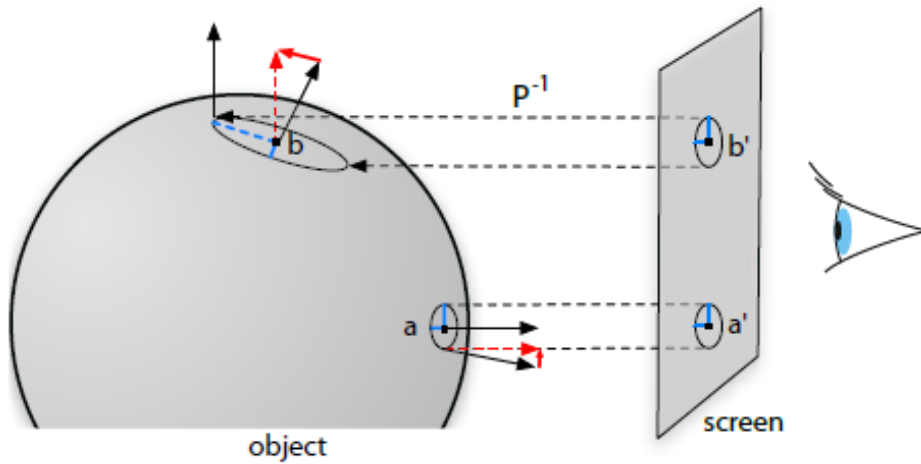


그림 8. 점 a' 보다 점 b' 에서 더 큰 최대 뷰-종속 곡률 [8]

Apparent ridges는 최대 뷰-종속 방향 t_1 에서 최대 뷰-종속 곡률 q_1 이 국소 최댓값을 갖는 점들의 집합이며 식 (5)로 정의한다.

$$D_{t_1} q_1 = 0 \quad (5)$$

뷰-종속 곡률은 contours에서 무한대에 근접하기 때문에 contours는 Apparent ridges에 의해 추출된다. 그래서 apparent ridges는 contours의 특성을 포함한다. 또한 apparent ridges는 $\kappa_1 > 0$ 인 부분에서 ridges와 같은 특성을 갖으며 $\kappa_1 < 0$ 인 부분에서 valleys와 와 같은 특성을 갖는다[8].

B. Autoencoder

머신러닝은 크게 지도학습(Supervised learning)과 비지도 학습 모델(Unsupervised learning)로 나눌 수 있다. 지도학습은 학습 데이터가 입력되면 이에 맞는 결과를 알려주며 학습하는 방식이며 비지도 학습은 입력 데이터만 주어진 상태로 데이터의 특징을 찾아내는 학습 방법이다. Autoencoder는 비지도 학습 모델 중 하나로서 출력 값이 입력 값과 근사하도록 학습하는 인공신경망이다 [10].

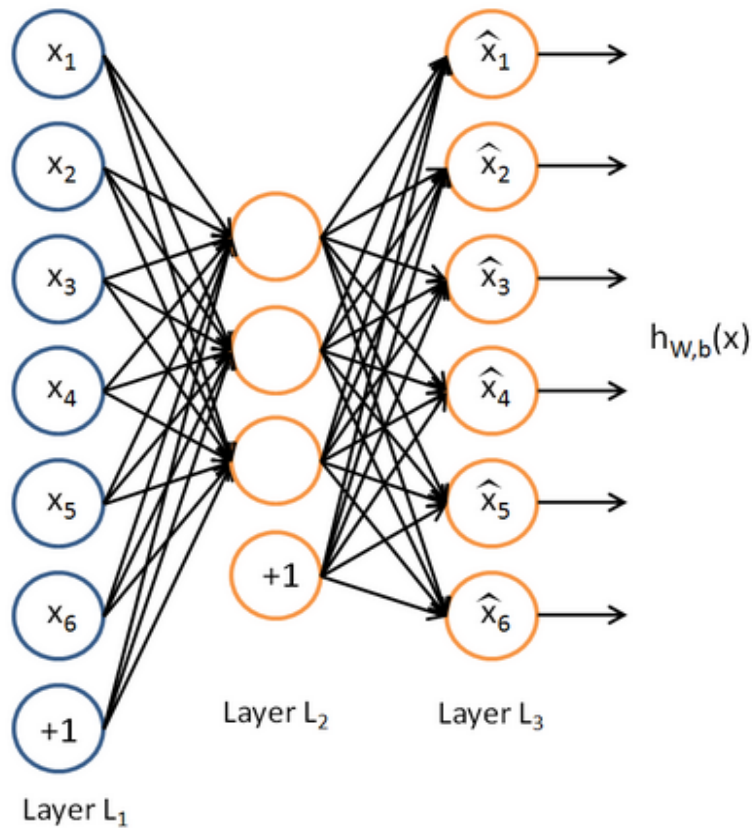


그림 9. Autoencoder의 구조 [11]

Autoencoder의 구조는 그림 9와 같으며 출력 값 $h_{W,b}(x)$ 가 입력 값 x 와 가까운 값이 되도록 학습하는 것이다. Autoencoder의 구조는 입력층(Input layer),

은닉층(Hidden layer), 출력층(Output layer) 3개의 계층으로 구성된다. 은닉층의 개수가 입력층보다 작은 것이 특징이고 데이터가 입력층에서 은닉층으로 이동하는 것을 encoding이라하고 은닉층에서 출력층으로 이동하는 것을 decoding이라 한다.

$$y = f_{\theta}(x) = s(Wx + b) \quad (7)$$

$$z = g_{\theta'}(x) = s(W'y + b') \quad (8)$$

입력 벡터 x 는 식 7을 통해 은닉 표현 y 으로 인코딩된다. 여기에서 s 는 활성화 함수를 나타내며 W 는 $d' \times d$ 차원의 가중치 행렬이고 b 는 편향(bias) 벡터이다. 은닉 계층 표현 y 는 식 8을 통하여 출력 벡터 z 로 디코딩된다.

Autoencoder는 입력과 출력의 차이를 줄이기 위해 손실 함수(Loss function)를 이용하여 최적화한다. z 가 연속적인 값일 때 MSE(Mean Square Error)를 사용하며 이진 분류일 때에는 CE(Cross Entropy)를 사용한다. 손실 함수의 값이 작을수록 오류가 적은 것을 뜻하며 출력 값이 입력 값에 근사하다는 것을 알 수 있다.

$$L_{MSE}(x, z) = \|x - z\|^2 \quad (9)$$

MSE는 입력 값과 출력 값의 차이를 제공한 것으로 식 9로 정의한다. 여기서 x 는 입력 값을 나타내며 z 는 출력 값을 나타낸다.

$$L_{CE}(x, z) = - \sum_{k=1}^d [x_k \log z_k + (1 - x_k) \log(1 - z_k)] \quad (10)$$

Cross Entropy는 식 10으로 정의한다. 여기서 x 는 입력 값, z 는 출력 값을 나타내고 k 는 k 번째 벡터 데이터를 의미하며 d 는 차원을 의미한다.

C. Convolutional Neural Network

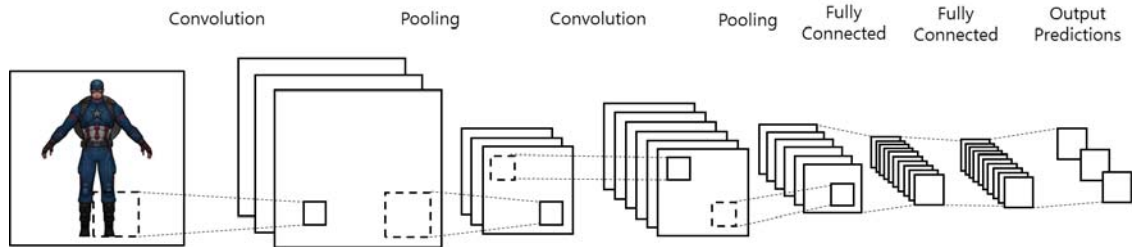


그림 10. Convolutional Neural Network의 기본 구조

Convolutional Neural Network(CNN)은 인간의 시신경 구조를 모방하여 만든 딥러닝 모델 중 하나이다. CNN의 기본구조는 그림 10과 같이 컨볼루션 계층(Convolution layer)과 풀링 계층(Pooling layer), 완전 연결 계층(Fully connected layer)로 구성된다.

컨볼루션 계층은 필터를 사용하여 입력데이터를 일정 간격으로 이동해가며 입력데이터를 연산한다. 그림 11은 입력데이터가 (6,6)이고 윈도우 크기가 (3,3)일 때 컨볼루션 계층에서 연산하는 예이다. 색칠된 부분이 윈도우(Window)크기이며 스트라이드(Stride)만큼 오른쪽으로 이동하며 필터(Filter)에 있는 컨볼루션 계층의 가중치와 연산한다. 이 연산의 출력 값을 은닉 계층으로 전달한다.

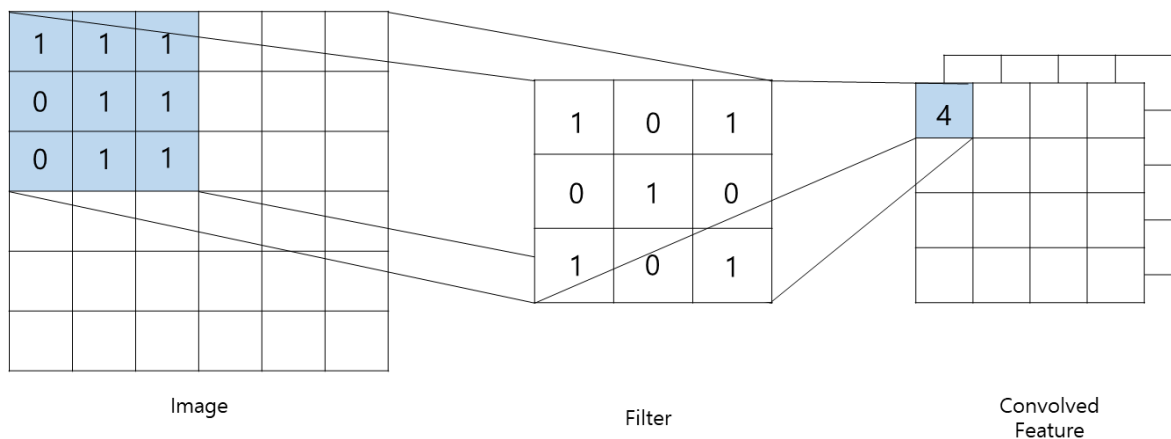


그림 11. 컨볼루션 계층의 연산 예

풀링 계층은 sub sampling으로 차원을 축소한다. 풀링의 종류로는 최대 풀링, 최소 풀링, 평균 풀링이 있다. 최대 풀링은 그림 12와 같이 윈도우안의 값들 중 최댓값을 특징 맵으로 만든다. 이와 비슷하게 최소 풀링은 최솟값, 평균 풀링은 평균값을 특징 맵으로 만든다. 이렇게 컨볼루션 계층과 풀링 계층이 반복된다보면 이전 계층과 모두 연결된 완전연결 계층이 된다.

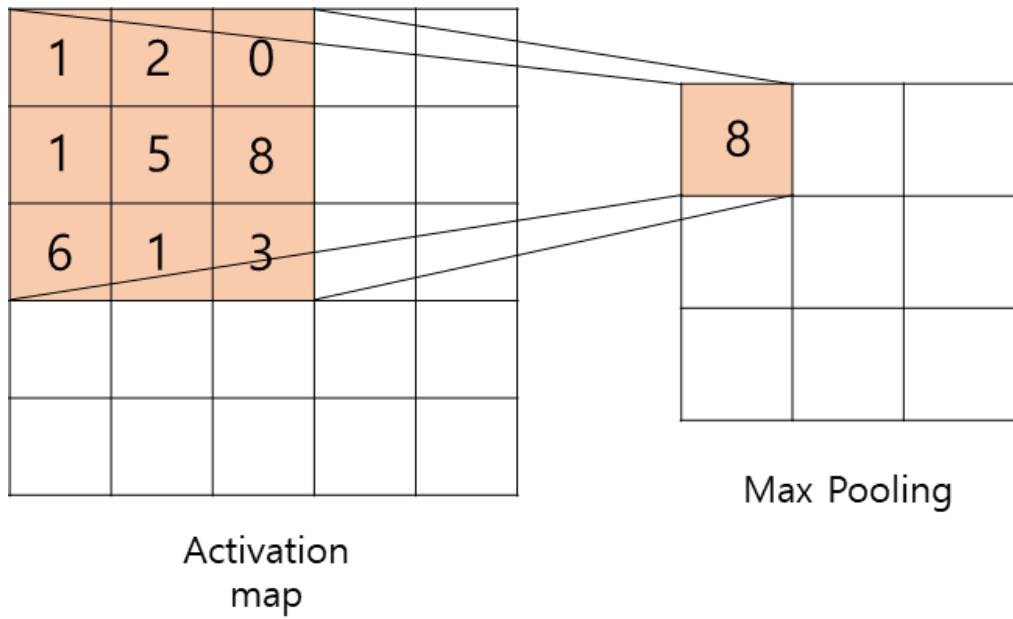


그림 12. 최대 풀링의 예

D. Generative Adversarial Network

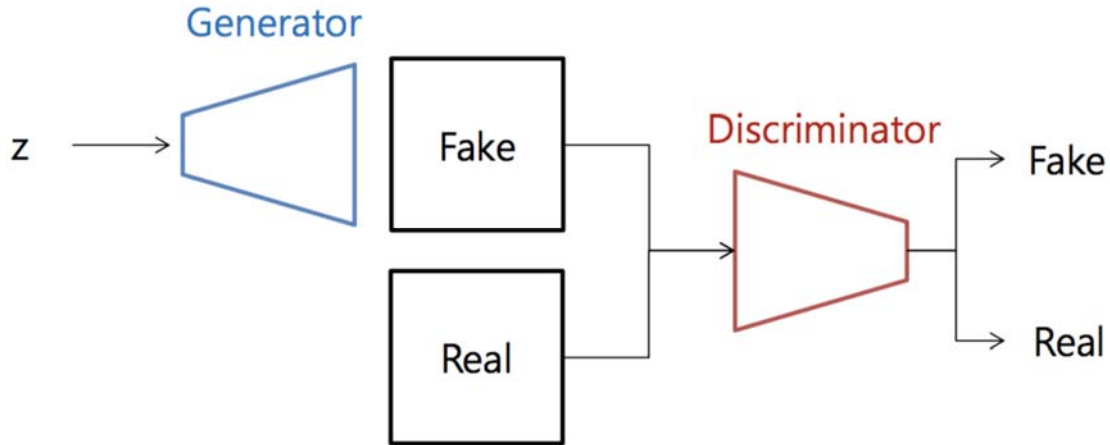


그림 13. Generative Adversarial Network의 기본구조

Generative Adversarial Network(GAN)는 적대적 생성 신경망으로 autoencoder와 같이 출력을 생성하는 모델이다. GAN의 기본구조는 그림 13과 같으며 생성자(Generator)와 구별자(Discriminator)라 불리는 두 가지 신경망의 경쟁을 통해 입력 데이터를 학습하고 출력을 생성한다. 생성자는 입력 데이터의 분포를 반영하여 학습하고 랜덤 벡터로부터 가짜 데이터를 생성하고 구별자는 생성자로부터 출력된 데이터가 진짜 데이터인지 가짜 데이터 인지 판별한다. GAN의 목적함수(Object Function)는 식 11과 같다[12].

$$\min_G \max_D V(D, G) = E_{x \sim p_{data}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log 1 - D(G(z))] \quad (11)$$

식 11에서 G 는 생성자, D 는 구별자를 뜻한다. x 는 입력 벡터이고 z 는 랜덤 벡터를 뜻한다. 구별자는 진짜 데이터와 가짜데이터를 구별할 수 있도록 식 11의 값이 큰 값을 갖도록 학습하며 생성자는 진짜 데이터와 비슷한 가짜 데이터를 생성하도록 식 11의 값이 작은 값을 갖게 학습한다.

GAN에 조건 항을 추가하여 데이터를 생성하는 Conditional Generative Adversarial Network(Conditional GAN)가 제안되었다. Conditional GAN은 특정 조건에 맞는 데이터를 생성할 수 있다. Conditional GAN의 목적함수는 식 12와 같이 GAN의 목적함수에 특정 조건을 나타내는 y 를 추가하여 나타낼 수 있다[13].

$$\min_G \max_D V(D, G) = E_{x \sim p_{data}(x)} [\log D(x|y)] + E_{z \sim p_z(z)} [\log (1 - D(G(z|y)))] \quad (12)$$

E. 드로잉 기반 모델링

NPR기술 중 하나인 드로잉 기반 모델링은 크게 두 가지로 분류할 수 있다. 첫 번째는 드로잉을 기반으로 3D 모델을 생성하는 방법이 있고 두 번째는 입력된 드로잉과 데이터베이스에서 3D 모델의 드로잉을 비교하여 가장 가까운 3D 모델을 검색하고 입력된 드로잉에 맞게 수정하는 방법이 있다[13, 14].

최근에는 딥러닝 기술을 드로잉 기반 모델링에 사용하는 연구가 진행되고 있다. 3D 모델을 생성하는 방법에서 딥러닝 기술은 2D line drawing을 입력하였을 때 3차원 정보를 나타내는 깊이(Depth) 맵과 법선(Normal)을 생성한다. 학습에 사용되는 데이터 셋을 생성하기 위해 3D 모델에서 특징선을 조합하여 앞면과 측면 line drawing을 추출한다. 추출된 line drawing을 딥러닝 모델에 학습시키고 전면과 측면 line drawing을 입력으로 주었을 때 깊이 맵과 법선 맵을 생성한다. 그리고 이 두 가지 이미지를 사용하여 점 구름(Point cloud)을 생성하고 최적화를 거쳐 메시를 생성한다. 그림 14는 line drawing을 입력하였을 때 3D 모델을 생성하는 전체 과정을 나타낸다[15].

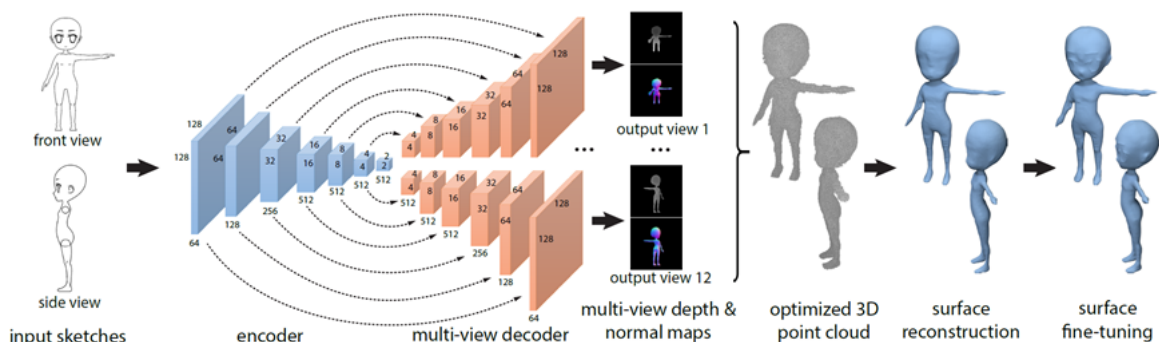


그림 14. 딥러닝 기술을 이용한 드로잉 기반 모델링 [15]

딥러닝을 사용하여 입력된 드로잉과 가장 비슷한 3D 모델을 찾는 방법은 스케치와 데이터베이스에 있는 3D 모델의 line drawing을 siamese network에 학습 시킨다. 그리고 스케치를 입력하였을 때 가장 비슷한 line drawing에 맞는 3D 모델을 찾는다[16].

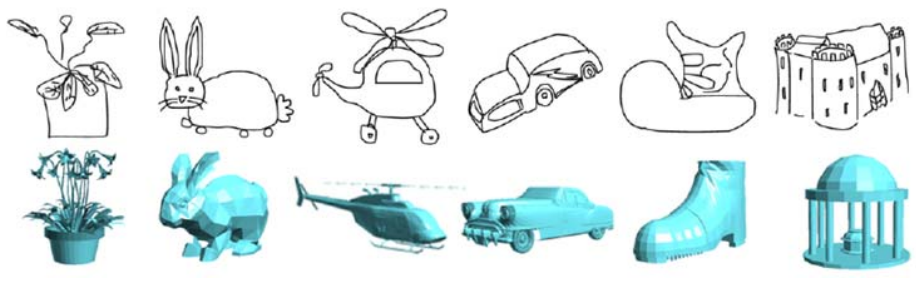
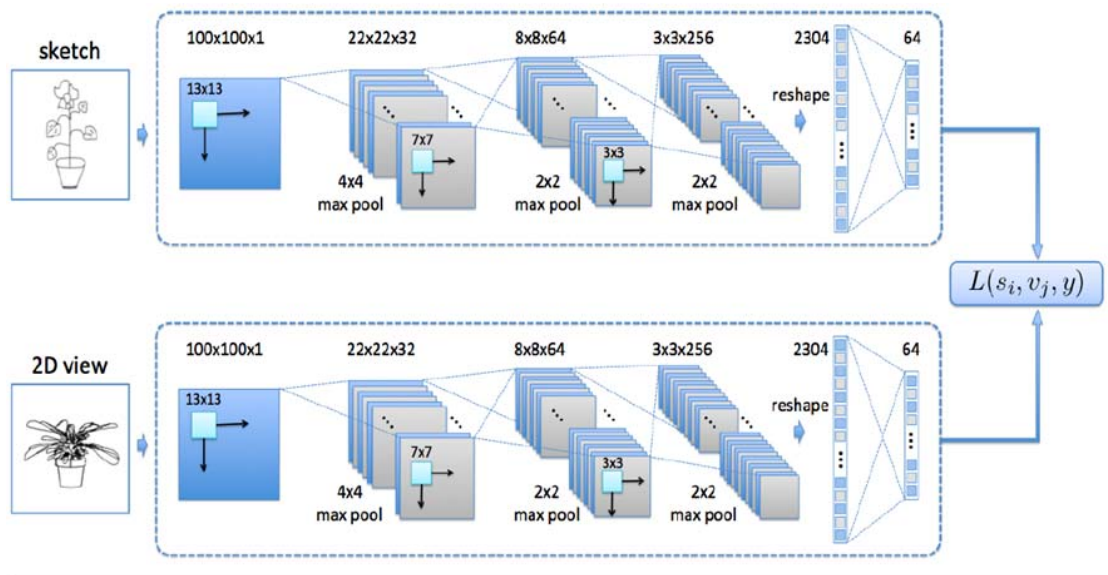


그림 15. 스케치기반 3D 모델 검색의 아키텍처(상)와 출력 예(하) [16]

III. Line drawing 추출, 검출, 생성

A. 3D 모델의 line drawing 추출

본 절에서는 앞서 설명한 다양한 특징선의 조합으로 line drawing 이미지를 생성한다. 여러 가지 특징선을 조합하여 데이터 셋을 생성한 이유는 각 특징선마다 모델의 형상을 나타내는 부분이 다르고 선을 통한 형상의 이해는 사람에 따라 다르기 때문이다. 특징선의 조합은 contours, suggestive contours, ridges, valleys, apparent ridges, boundaries로 구성 하였다.

투영 방식은 직교 투영(Orthographic projection)을 사용하여 측면, 정면, 후면 세 가지 시점에서 line drawing 이미지를 생성하였고(그림 16), 특징선 추출에 사용하는 3D 모델은 애니메이션, 게임 캐릭터의 3D 모델을 사용하였다.

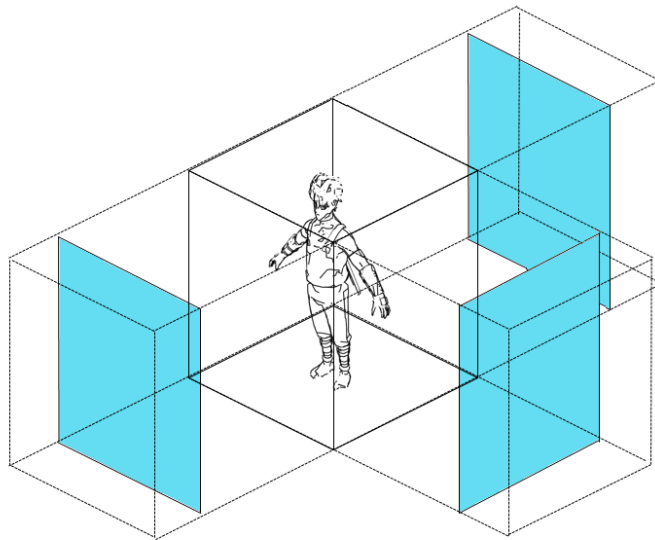


그림 16. 직교 투영 방식을 사용한 line drawing 이미지 생성

그림 17은 3D 캐릭터 모델에서 특징 선을 추출한 예이다. 그림 17의 (a)는 contours, ridges, valleys, boundaries를 사용한 line drawing이고 그림 17의 (b)는 contours, suggestive contours, ridges, valleys, apparent ridges, boundaries를 사용한 line drawing이다. 그림 17의 (a)와 (b)를 비교하면 팔, 코트, 복부 부분에서 (b)가 (a)보다 기하학적 특성을 잘 표현하는 것을 볼 수 있다. 그림 17 (a)와 (b)에서 점선 원으로 표시된 팔 끝부분과 다리 안쪽에서 선이 그려지지 않는데 이는 해당 부분의 정점들이 특징선으로 검출 되지 않기 때문이다.

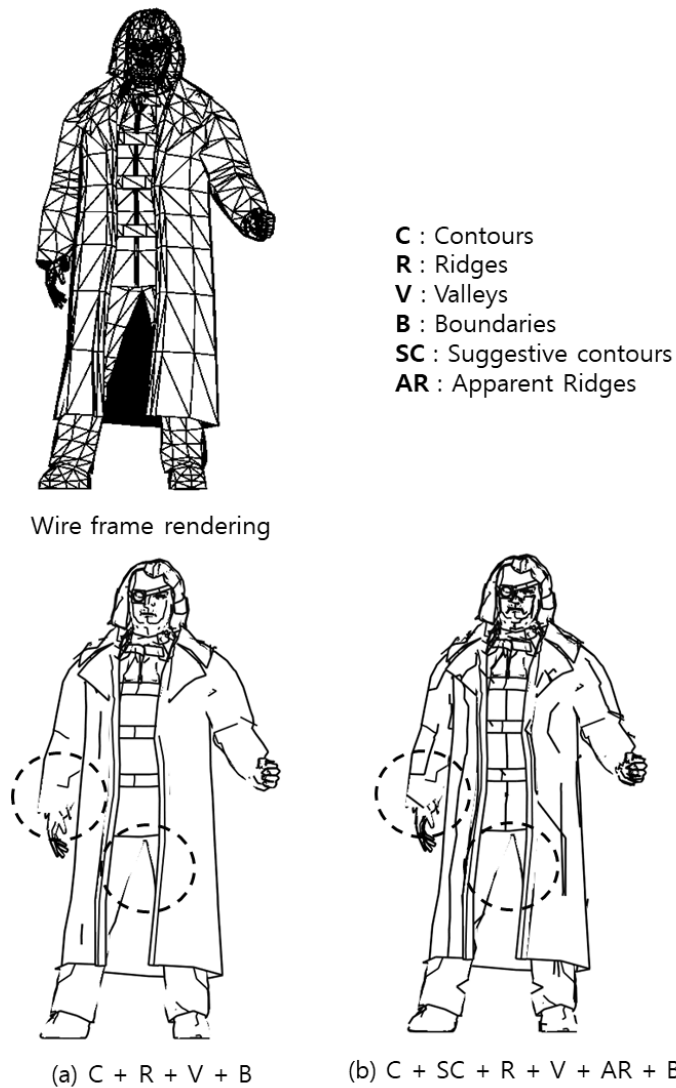


그림 17. 특징선의 조합으로 나타낸 line drawing

이를 해결하기 위해 특징선 추출 전 3D 모델에 대해 subdivision surface를 수행하였다[17]. 그림 18에서 subdivision surface의 결과로 3D 캐릭터 모델의 vertex의 수가 증가하고 표면이 매끄러워지는 것을 확인할 수 있다. 연구에 사용하는 3D 캐릭터 모델을 예로 그림 19의 초기상태와 subdivision surface를 1회, 2회 적용한 메시의 곡률을 비교하면 초기 상태보다 머리, 가슴, 허벅지 등 곡률이 증가함을 알 수 있다.

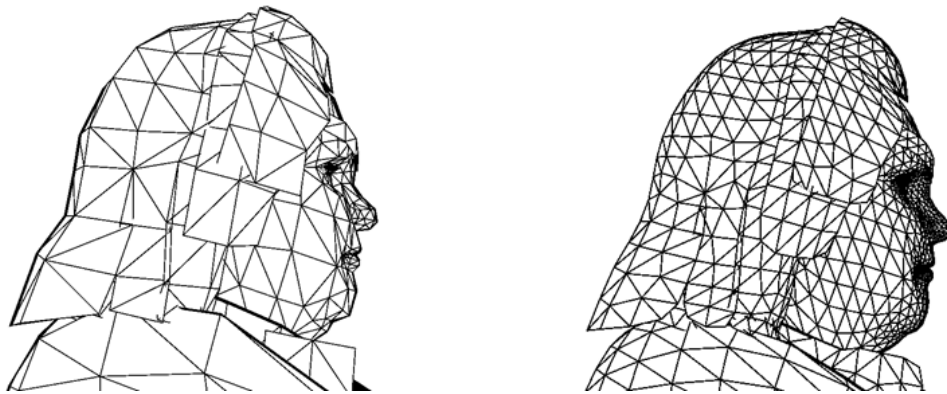


그림 18. 3D 캐릭터 모델의 subdivision surface 전(좌) · 후(우) 비교

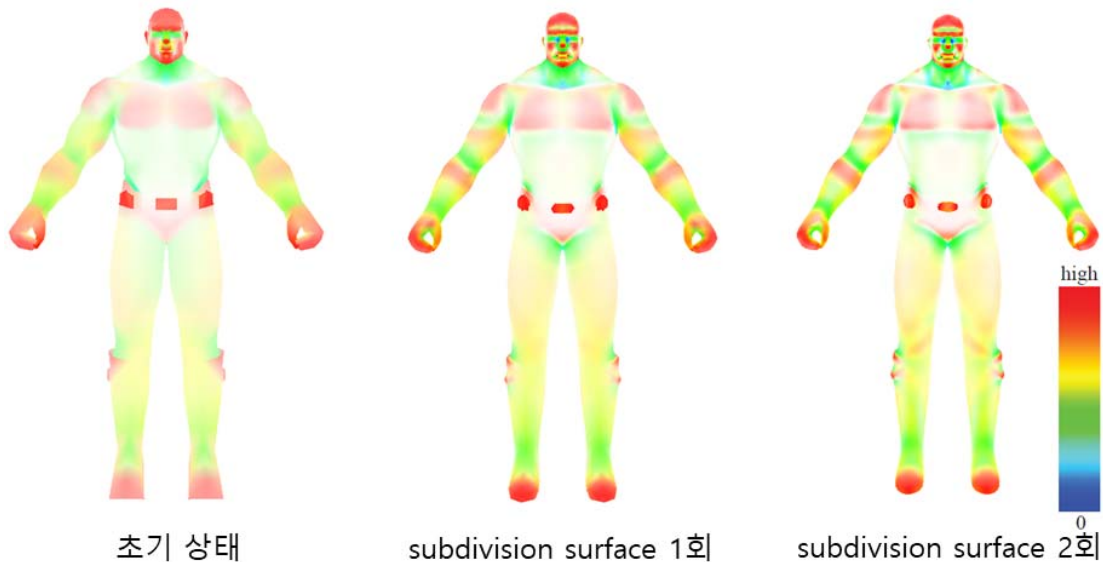


그림 19. Subdivision surface 횟수에 따른 곡률의 변화

그림 20은 subdivision surface의 횡수에 따른 line 조합의 결과이다. subdivision surface의 횡수에 따라 비교하면 그림 20 (b)에서 코트의 형상 부분(점선 사각형)에 선이 없지만 그림 20 (e)의 코트(점선 사각형)는 곡률의 변화에 따라 선이 그려져 코트의 굴곡을 인지할 수 있다. 또한 얼굴이나 신발 부분에도 선이 그려져 형상을 인지하는데 도움을 준다.

특징선의 조합에 따라 비교하면 그림 20 (b)의 팔 부분(실선 사각형)에는 선이 없지만 그림 20 (c)의 팔 부분(실선 사각형)에 선이 그려져 팔이 구부러져 있다는 것을 인지할 수 있다. 그러나 얼굴 부분에서는 오히려 suggestive contours와 apparent ridges가 추가되어 형상을 알아보기 힘들게 되었다. 그 이유는 얼굴을 3D 모델로 제작 할 때 눈, 코, 입은 굴곡이 많아 표현되는 특징선이 많기 때문이다. 그래서 얼굴 부분은 그림 20 (f)에서 볼 수 있듯이 subdivision surface를 수행할수록 오히려 형상을 알아보기 힘들어진다.

따라서 학습에 사용할 데이터 셋은 최대한 많은 선을 표현하기 위해 3D 모델에 subdivision surface를 2회 수행하고 특징선은 contours, suggestive contours, ridges, valleys, apparent ridges, boundaries 조합을 사용한다. 특징선의 조합으로 line drawing 이미지를 측면, 앞면, 뒷면 3가지 방향에서 생성하여 이를 데이터 셋으로 사용한다.

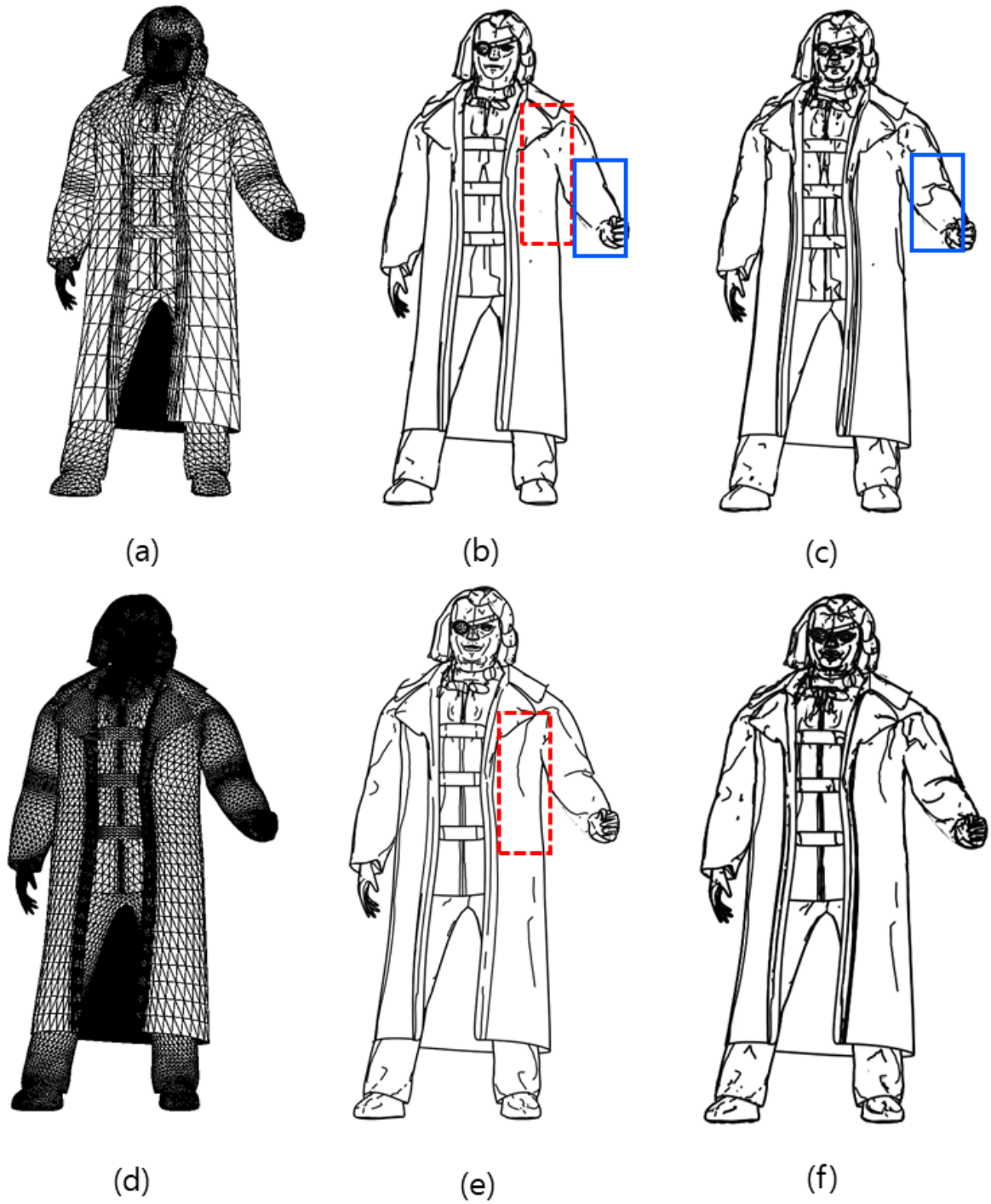


그림 20. Subdivision surface 횟수에 따른 3D 모델의 특징선 추출
 (a) 1회, wire frame (b) 1회, 첫 번째 라인 조합 (c) 1회, 두 번째 라인 조합
 (d) 2회, wire frame (e) 1회, 첫 번째 라인 조합 (f) 2회, 두 번째 라인 조합

B. Line drawing 검출 및 생성

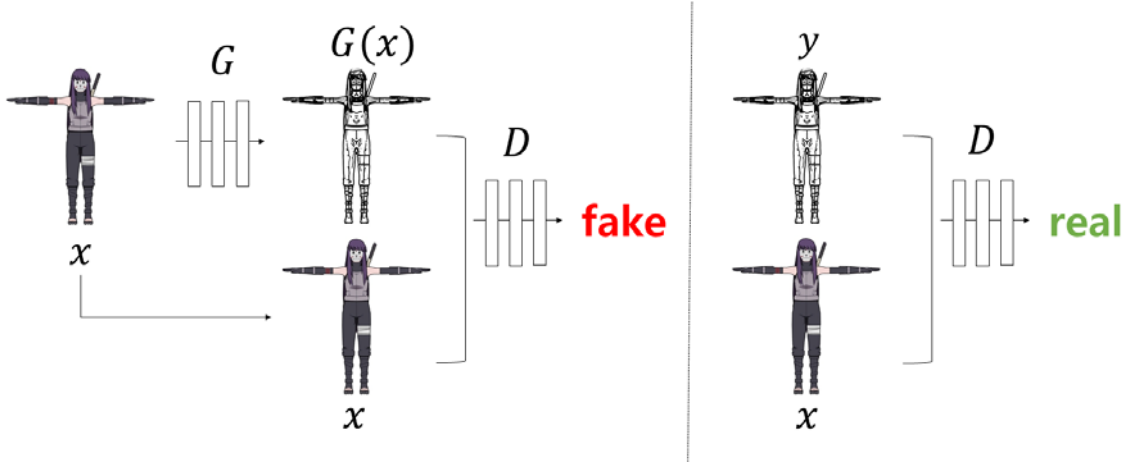


그림 21. 2D 카툰이미지에서 line drawing 생성을 위한 Conditional GAN 학습

특징선 검출과 생성은 Conditional GAN을 사용하는 pix2pix[19]를 수정하여 학습을 수행한다. pix2pix는 이미지에서 이미지를 생성하는 네트워크로서 기존 pix2pix는 에지(Edge) 이미지에서 컬러 이미지를 생성하는데 본 연구에 맞게 카툰 이미지에서 line drawing을 검출하도록 수정하였다.

Conditional GAN은 생성자 네트워크와 구별자 네트워크로 구성된다. Conditional GAN의 학습은 두 개의 네트워크를 학습하게 되는데 먼저 convolution과 deconvolution으로 이루어진 생성자 네트워크를 학습한다. 생성자는 각각 8개의 convolution, deconvolution 계층으로 이루어져 있으며 U-net Network를 사용한다. encoding과정(Convolution)에서는 이미지를 이해하기 위해 특징 맵의 크기를 줄이면서 핵심적 특징들을 검출한다. Encoding이 최종적으로 끝나고 나온 특징 맵을 바탕으로 decoding(Deconvolution)을 통해 toon data를 line drawing data로 최종적으로 복원하게 된다. 이 과정에서 손실된 정보를 보강하기 위해 skip connection을 사용한다. 구별자는 입력된 이미지를 기반으로 하여 patch를 만들어내고 이 patch를 가지고 입력이 지 판별해낸다. 구별자는 Patch GAN을 사용하여 생성자가 만들어낸 이미지가 입력된 이미지에 상응하는 데이터인지 판별한다. 전체 이미지의 진짜와 가짜의 여부를 판단하는 것이 아닌 특정 크기의 patch단위로 판별하고, 그 결과에서 평균을 취하는 방식이다[19].

본 연구에서 입력데이터는 2D 카툰이미지 x 와 이에 해당하는 line drawing y 로 이루어진다. 생성자 G 는 그림 21과 같이 입력된 2D 카툰이미지를 학습하고 line drawing과 비슷한 가짜 데이터 $G(x)$ 를 생성한다. 구별자 D 는 2D 카툰이미지 x 와 진짜 데이터 y , 생성자로부터 만들어진 가짜 데이터 $G(x)$ 를 번갈아가며 학습하고 y 와 $G(x)$ 가 진짜 데이터인지 아닌지 판별한다. 식 13은 본 연구에서 사용한 Conditional GAN의 목적 함수이다.

$$V(G, D) = \min_G \max_D E_y [\log(D(y))] + E_x [\log D(1 - D(G(x)))] + E_{x,y} [\|y - G(x)\|_1] \quad (13)$$

식 13에서 볼 수 있듯이 생성자는 입력된 2D 카툰 이미지와 최대한 비슷한 이미지를 생성해야 하므로 목적함수의 결과 값이 최솟값을 갖도록 학습하며, 구별자는 생성자가 만들어낸 line drawing이 2D 카툰이미지에 상응하는 데이터인지 판별하기 위해 목적 함수의 값이 최댓값을 갖는다. 목적 함수의 값을 토대로 최적화하여 학습을 반복한다.

구별자는 학습초기에는 생성자로부터 생성된 line drawing이 2D 카툰 이미지에 상응하는 데이터 인지 구분할 수 있지만 학습이 진행될수록 생성자는 y 에 가까운 $G(x)$ 를 생성하므로 어떤 것이 진짜인지 가짜인지 구분하기 힘들어진다. 이러한 반복 학습으로 2D 카툰이미지를 입력하면 실제 line drawing에 가까운 데이터를 생성할 수 있다.

IV. 실험 및 결과 분석

A. 실험 환경

표 1. 실험 환경 및 설계

사용 모델	Conditional GAN
실험 환경	CPU i7-8700K, GPU-1080ti 11GB
트레이닝 셋	정면 816장
발리데이션 셋	정면 80장
학습 횟수	200회 ~ 400회 (100단위 증가)
학습시간	100회 당 약 6시간
특징선	contours, suggestive contours, ridges, valleys, apparent ridges, boundaries

본 연구에서 제안하는 방법을 실험하기 위한 환경과 실험 설계는 표 1과 같다. 시스템은 CPU는 i7-8700K, GPU는 1080ti 11GB으로 구성되며 학습에 사용한 데이터 셋은 816개의 3D 모델에서 생성한 카툰 이미지(그림 22 좌)와 line drawing (그림 22 우)으로 구성되며 이미지의 크기는 512 x 512 이다. 학습에 사용한 line drawing은 3D 모델에 subdivision surface를 2회 수행하고 특징선은 contours, suggestive contours, ridges, valleys, apparent ridges, boundaries 조합으로 구성된다. 또한 subdivision surface가 어떤 영향을 미치는지 알아보기 위해 subdivision surface를 수행하지 않은 데이터 셋으로 실험을 추가적으로 수행하였다.

테스트에 사용한 발리데이션 셋은 데이터 셋과 같이 80개의 3D 모델에서 추출한 카툰 이미지와 line drawing으로 구성된다. 테스트는 Conditional GAN 모델을 200회 부터 400회까지 100단위씩 증가하여 학습시킨 후 발리데이션 셋을 입력하여 결과를 확인하였다.



그림 22. 학습에 사용되는 카툰 이미지(좌)와 line drawing(우)

B. 실험 결과 분석

본 연구의 실험결과를 시각적으로 분석하면 2D 카툰이미지를 입력하였을 때 입력데이터에서 보이지 않는 3D 모델의 기하학적 속성이 출력된 line drawing에 나타나는 것을 확인 할 수 있다. 그림 23에서 입력된 2D 카툰 이미지의 팔과 허벅지, 복부 부분은 표면의 굴곡을 시각적으로 인지하기 힘들다. 하지만 출력된 line drawing은 표면의 굴곡이 선으로 표현되어 굴곡을 시각적으로 인지할 수 있다.

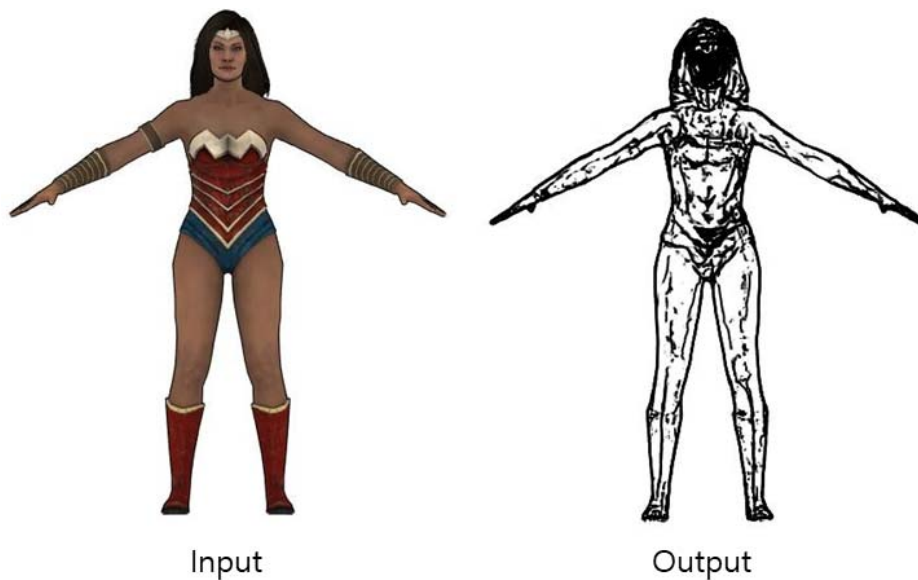


그림 23. 2D 이미지에서 기하학적 속성을 나타내는 line drawing 추출

3D 모델의 subdivision surface 수행에 따른 결과를 시각적으로 분석하면 표 1에서 볼 수 있듯이 수행하지 않았을 때보다 2회 수행한 결과가 기하학적 속성을 잘 나타내었다. 표 2는 subdivision surface을 수행하지 않은 데이터 셋과 2회 수행한 데이터 셋으로 학습하고 출력한 결과이다. subdivision surface을 2회 수행한 결과가 가슴, 복부, 다리 부분에 수행 안한 결과보다 시각적으로 기하학적 속성을 잘 나타내는 것을 확인할 수 있다. 이러한 결과가 나온 이유는 subdivision surface을 수행함으로써 3D 모델의 굴곡이 증가하고 결과적으로 표현되는 특징선이 많기 때문으로 보인다. 표 3 ~ 표 5는 2D 카툰 이미지에서 line drawing을 추출한 결과이다.

표 2. Subdivision surface 수행 횟수에 따른 결과



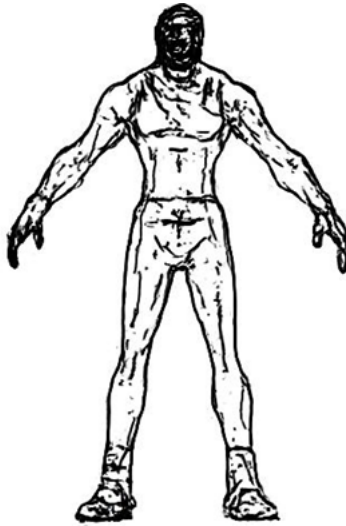
		
<p>Input</p>	<p>Output, Epoch = 200, Sub = 0</p>	<p>Output, Epoch = 200, Sub = 2</p>

표 3. Conditional GAN을 이용한 line drawing 추출 결과 - 1


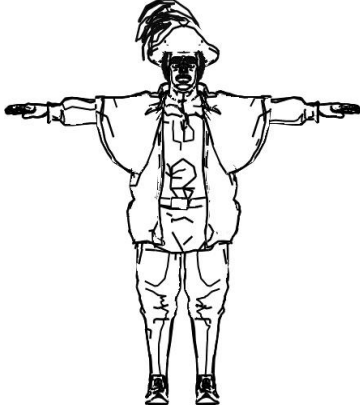
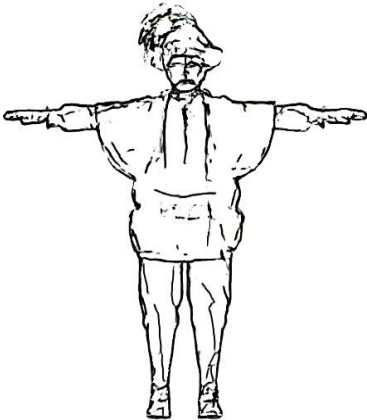
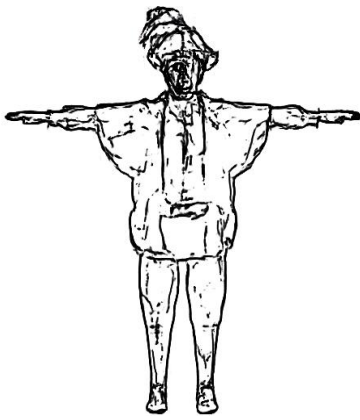
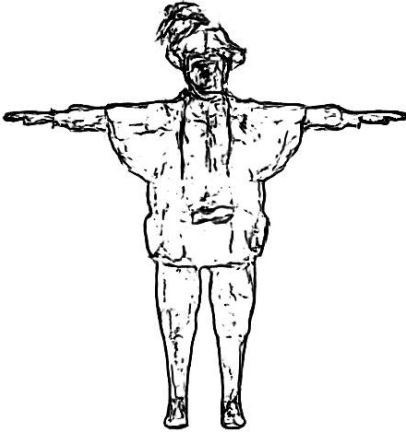
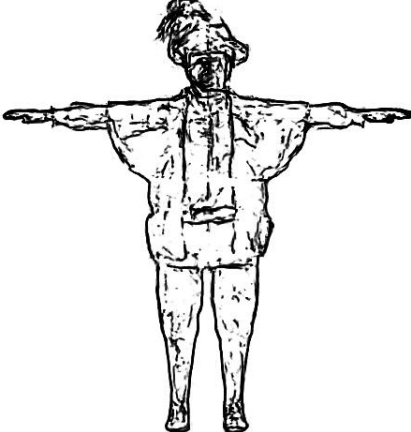
	
<p>Input</p>	<p>Ground Truth</p>
	
<p>Output, Epoch = 200, Sub = 0</p>	<p>Output, Epoch = 200, Sub = 2</p>
	
<p>Output, Epoch 300, Sub = 2</p>	<p>Output, Epoch 400, Sub = 2</p>

표 4. Conditional GAN을 이용한 line drawing 추출 결과 - 2


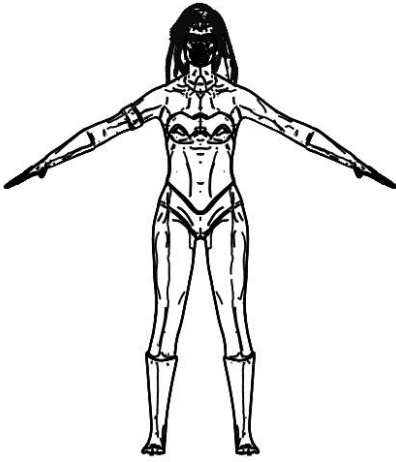
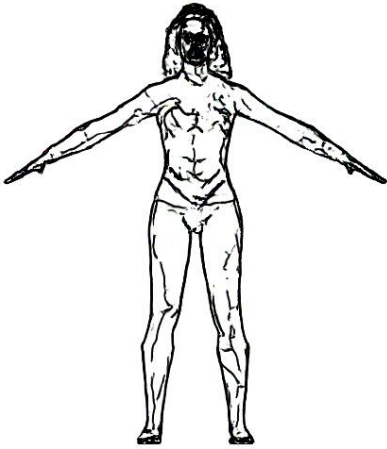
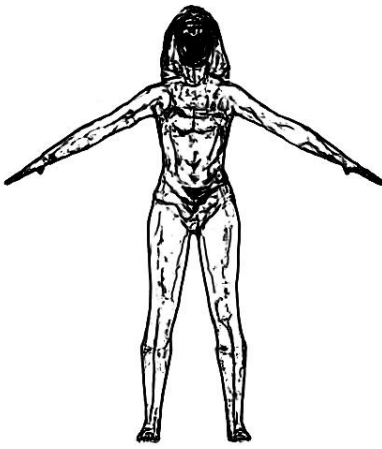
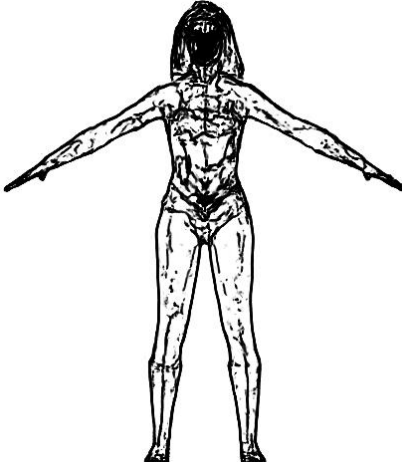
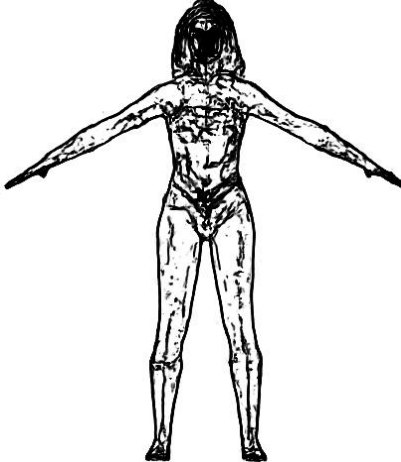

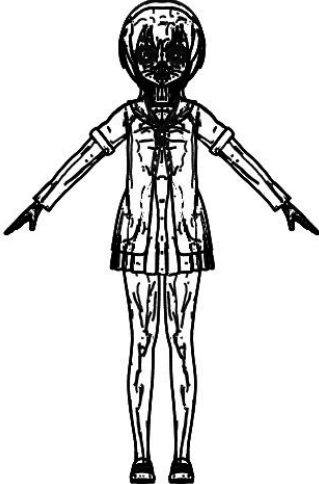
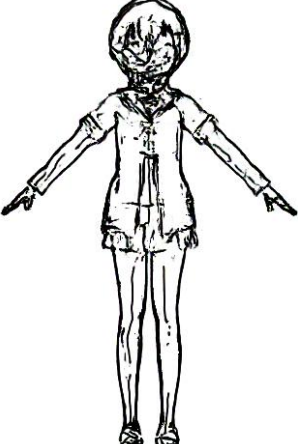
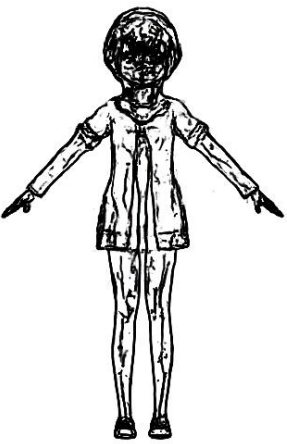
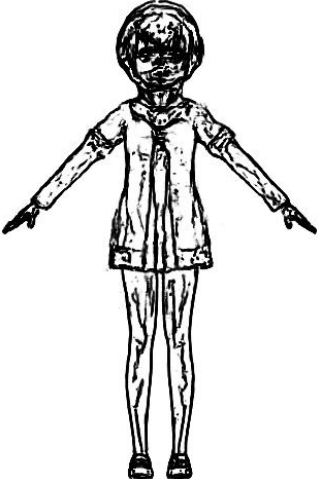
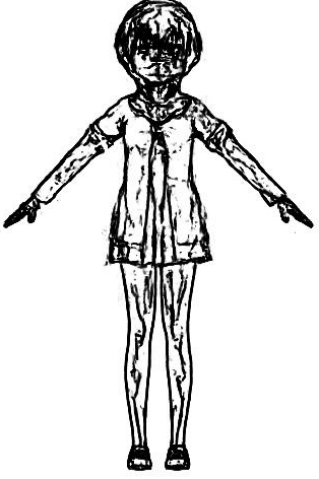
	
<p>Input</p>	<p>Ground Truth</p>
	
<p>Output, Epoch = 200, Sub = 0</p>	<p>Output, Epoch = 200, Sub = 2</p>
	
<p>Output, Epoch 300, Sub = 2</p>	<p>Output, Epoch 400, Sub = 2</p>

표 5. Conditional GAN을 이용한 line drawing 추출 결과 - 3

	
<p>Input</p>	<p>Ground Truth</p>
	
<p>Output, Epoch = 200, Sub = 0</p>	<p>Output, Epoch = 200, Sub = 2</p>
	
<p>Output, Epoch 300, Sub = 2</p>	<p>Output, Epoch 400, Sub = 2</p>

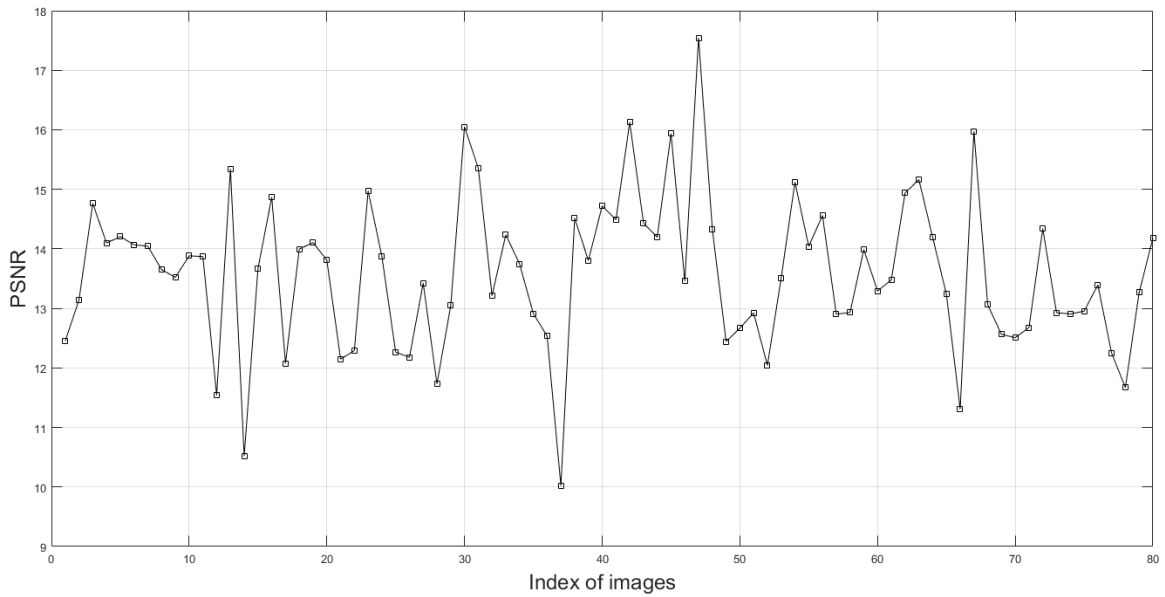


그림 24. Ground Truth와 Output의 PSNR 그래프

본 연구의 실험 결과는 Conditional GAN 모델을 사용하여 2D 카툰 이미지에서 3D 모델의 기하학적 특성을 보이는 line drawing을 생성하므로 일반적인 2D 이미지를 생성하는 연구와 비교할 수 없다. 따라서 정량적으로 측정하기 위해 PSNR(Peak Signal-to-noise ratio)를 사용하였다. PSNR은 최대 신호 대 잡음비로 본 연구에서는 Output이 Ground Truth에 얼마만큼 가깝게 생성되었는지 수치적으로 나타낸다. 그림 24는 발리데이션 셋 각각의 데이터(x 축)에 대한 PSNR (y 축)을 나타내는 그래프이다. PSNR 측정 결과 최솟값은 약 10, 최댓값 약 17의 결과를 보였다. PSNR의 측정 결과 트레이닝 셋과 발리데이션 셋의 일관성이 부족하여 측정값이 일정치 않은 것으로 보인다. 또한 트레이닝 셋에 공통된 특성을 학습하므로 발리데이션 셋에 이와 비슷한 데이터가 있다면 Output이 Ground truth에 가까운 결과를 나타내고 그렇지 않다면 그리지 못하는 것을 알 수 있다.

V. 결 론

본 연구에서는 Conditional GAN 모델을 사용하여 2D 카툰원화에서 line drawing을 자동으로 추출하는 방법을 제안하였다. 제안한 방법은 데이터 셋을 생성하기 위해 3D 모델을 subdivision surface를 2회 수행하고 contours, suggestive contours, ridges, valleys, apparent ridges, boundaries의 조합으로 line drawing을 추출하였다. 그리고 생성된 데이터 셋을 Conditional GAN 모델에 2D 카툰 이미지와 line drawing을 이용하여 학습시켰다. 마지막으로 테스트는 학습 횟수를 처음 100회에서 400회까지 100단위로 증가시키며 학습시키고, 그 후 학습이 끝나면 발리데이션 셋을 이용하여 결과를 확인하였다.

실험 결과 subdivision surface 수행 하지 않았을 때와 수행했을 때를 비교하면 수행 했을 때 형상을 나타내는 선이 더 많이 그려지는 것을 확인할 수 있었으며, 학습 횟수에 따라 비교하면 학습 횟수가 증가할수록 선의 표현이 많아지고 뚜렷하게 그려지는 결과를 보였다. 결과적으로 2D 카툰원화에서 보이지 않던 기하학적 속성이 본 연구의 실험결과 line drawing으로 표현되는 것을 확인할 수 있었다.

본 연구의 제한점으로는 데이터 셋을 생성하는데 사용되는 3D 모델의 일관성 문제이다. 본 연구에 사용되는 3D 모델은 정밀도를 나타내는 polygon의 수가 서로 달라 line drawing의 추출에 있어 일관성이 부족하였다. 다시 말하면 low poly로 제작된 3D 모델에서는 특징선이 검출되지 않아 subdivision surface를 수행하여 특징선을 검출하였다. 반대로 high poly로 제작된 3D 모델은 subdivision surface를 수행하면 특징선 검출이 과하게 되는 역효과가 발생되었다. 따라서 subdivision surface의 수행 횟수를 기준으로 line drawing을 추출하였다.

향후 연구로는 실제 카툰 원화를 이용하여 실험을 수행하고 본 연구의 결과로 생성된 line drawing을 드로잉 기반 모델링에 사용하여 2D 카툰 이미지에서 3D 모델을 생성하도록 하겠다. 본 연구의 결과를 3D모델 생성에 입력으로 사용하면 contour와 suggestive contour를 입력으로 사용하는 기존 연구보다 캐릭터의 형상이 자세하게 표현된 3D 모델을 얻을 수 있을 것으로 예상된다.

참고문헌

- [1] 김성예, et al. "비사실적 렌더링 기술동향." *한국멀티미디어학회지* 9.3 · 4 (2005): 78-94.
- [2] Kang, Henry, Seungyong Lee, and Charles K. Chui. "Coherent line drawing." *Proceedings of the 5th international symposium on Non-photorealistic animation and rendering*. ACM, 2007.
- [3] Rusinkiewicz, Szymon, et al. "Line drawings from 3D models." *ACM SIGGRAPH 2008 classes*. ACM, 2008.
- [4] Decaudin, Philippe. "Cartoon-looking rendering of 3D-scenes." *Syntim Project Inria 6* (1996).
- [5] Lake, Adam, et al. "Stylized rendering techniques for scalable real-time 3d animation." *Proceedings of the 1st international symposium on Non-photorealistic animation and rendering*. ACM, 2000.
- [6] DeCarlo, Doug, et al. "Suggestive contours for conveying shape." *ACM Transactions on Graphics (TOG)* 22.3 (2003): 848-855.
- [7] DeCarlo, Doug, Adam Finkelstein, and Szymon Rusinkiewicz. "Interactive rendering of suggestive contours with temporal coherence." *Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering*. ACM, 2004.
- [8] Judd, Tilke, Frédo Durand, and Edward Adelson. "Apparent ridges for line drawing." *ACM Transactions on Graphics (TOG)*. Vol. 26. No. 3. ACM, 2007.
- [9] Bender, J., et al. "Comparative evaluation of feature line techniques for shape depiction." (2014).
- [10] Masci, Jonathan, et al. "Stacked convolutional auto-encoders for hierarchical feature extraction." *International Conference on Artificial Neural Networks*. Springer, Berlin, Heidelberg, 2011.
- [11] Ng, Andrew. "Sparse autoencoder. 2011." URL <https://web.stanford.edu/class/cs294a/sparseAutoencoder.pdf>.
- [12] Goodfellow, Ian, et al. "Generative adversarial nets," *Advances in neural information processing systems* pp. 2672-2680, 2014.

- [13] Olsen, Luke, et al. "Sketch-based modeling: A survey." *Computers & Graphics* 33.1 (2009): 85-103.
- [14] Kazmi, Ismail Khalid, et al. "Efficient sketch-based creation of detailed character models through data-driven mesh deformations." *Computer Animation and Virtual Worlds* 26.3-4 (2015): 469-481.
- [15] Lun, Zhaoliang, et al. "3D shape reconstruction from sketches via multi-view convolutional networks." *3D Vision (3DV), 2017 International Conference on*. IEEE, 2017.
- [16] Wang, Fang, Le Kang, and Yi Li. "Sketch-based 3d shape retrieval using convolutional neural networks." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015.
- [17] Zorin, Dennis, et al. "Subdivision for modeling and animation. SIGGRAPH 2000 Course Notes, 2000. "
- [18] Isola, Phillip, et al. "Image-to-image translation with conditional adversarial networks." *arXiv preprint* (2017).
- [19] Mirza, M, and Simon Osindero. "Conditional generative adversarial nets," *arXivpreprint arXiv:1411.1784*, 2014.