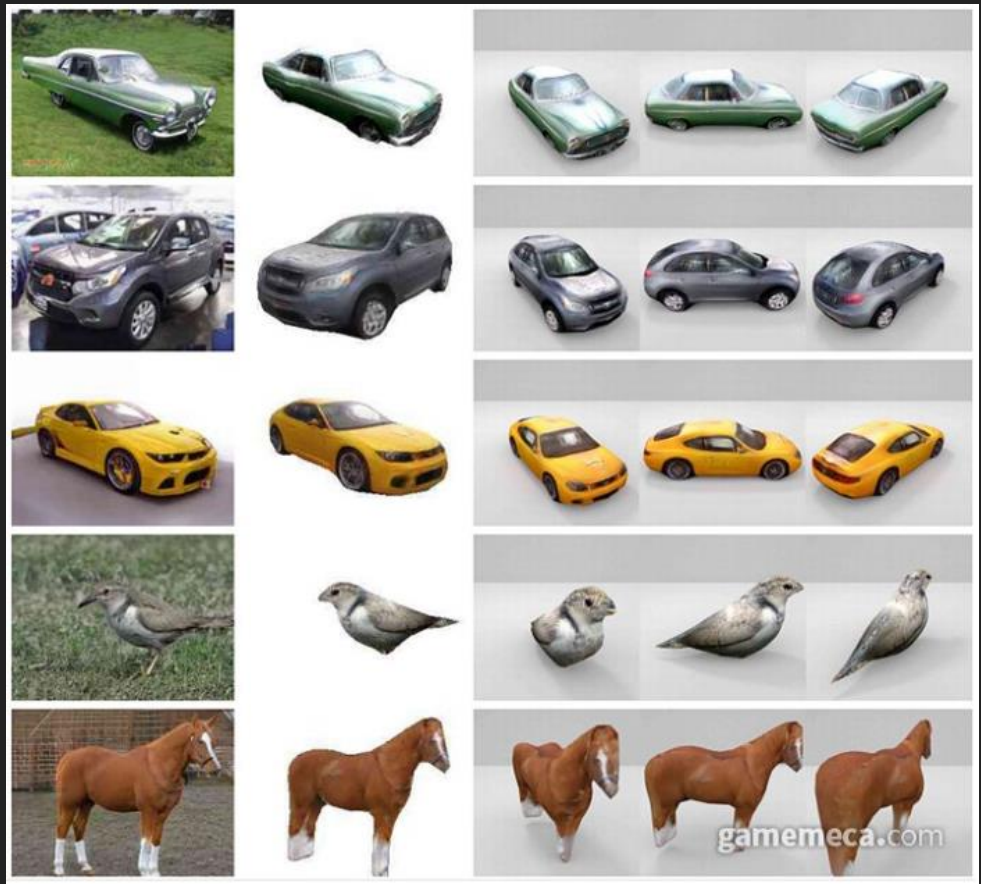
2D 사진을 3D 모델링으로, 엔비디아 'GANverse3D' 공개



▲ 사진 한 장만으로 손쉽게 3d 모델링을 만들 수 있다 (사진출처: 엔비디아 기술 논문)

엔비디아는 자사의 AI 리서치 랩(AI Research Lab)에서 생성적 적대 신경망 기반으로 개발한 'GANverse3D' 애플리케이션을 지난해 10월 공개했었다. 이는 평면 이미지를 사실적인 3D 모델로 변환해 가상 환경에서 시각화 및 제어를 가능하게 한다.

이는 특정 자동차 사진 한 장만으로, 사실적인 전조등, 후미등, 점멸등까지 완비한 3D 모델을 구축해 가상의 장면에서 주행하게 만들 수 있다.

20182788 고민성

Image GANs meet Differentiable Rendering for Inverse Graphics and Interpretable 3D Neural Rendering

Yuxuan Zhang*, Wenzheng Chen*, Huan ling, Jun Gao, Yinan Zhang, Antonio Torralba, Sanja

논문의 요약

- 객체의 이미지 기반 3D 재구성을 위해 Style-GAN network와 DIB-R 결합

- Style-GAN network를 통해 데이터를 생성하고 differentiable renderers(DIB-R)을 통해 객체 속성을 추출해낸다.

- 추출해낸 객체 속성을 이용하여 새로운 객체를 생성할 수 있다.

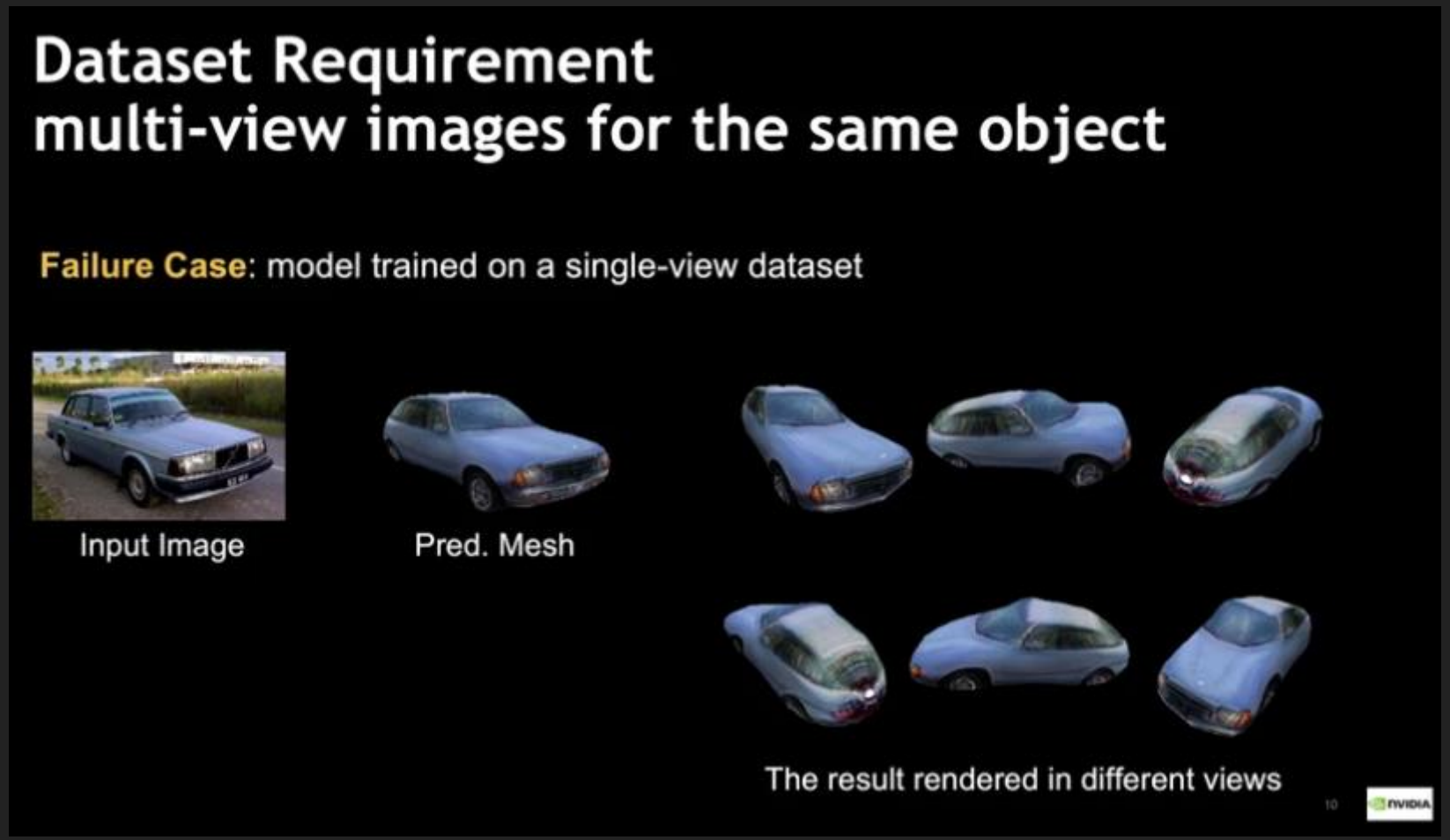
Introduction

2차원 사진에서 기하학, 질감, 재료 및 빛과 같은 속성을 추론하여 3차원으로 만드는 것은 AR/VR, 로봇 공학, 아키텍쳐 등 여러 분야의 핵심이고 최근 몇 년 간 관심이 대단하였습니다.

2차원 이미지에서 3D로 이동하는 과정을 "Inverse graphics"라 하는데 이를 위해서는 다양한 시점의 데이터가 있어야 좋은 성능을 나타낼 수 있습니다.

하지만, 동일한 객체를 서로 다른 각도에서 캡쳐하는 데이터셋은 현실적으로 드뭅니다. 또한 동일 객체의 여러 시점에 대한 데이터셋을 사람이 직접 확보하는데는 많은 어려움이 수반됩니다. 그리고 유명한 데이터셋이 몇몇 있지만 이는 도메인의 차이로 어려움을 겪는 상황이 발생합니다.

한 시점으로 3D를 만들어냈을 때의 결과



2D 이미지에는 자동
차의 뒷 부분과 위에
부분에 대해 나타나
있지 않기 때문에 3D
데이터 상에서 뭉개져
있는 현상이 나타났다.

STYLEGAN AS SYNTHETIC DATA GENERATOR

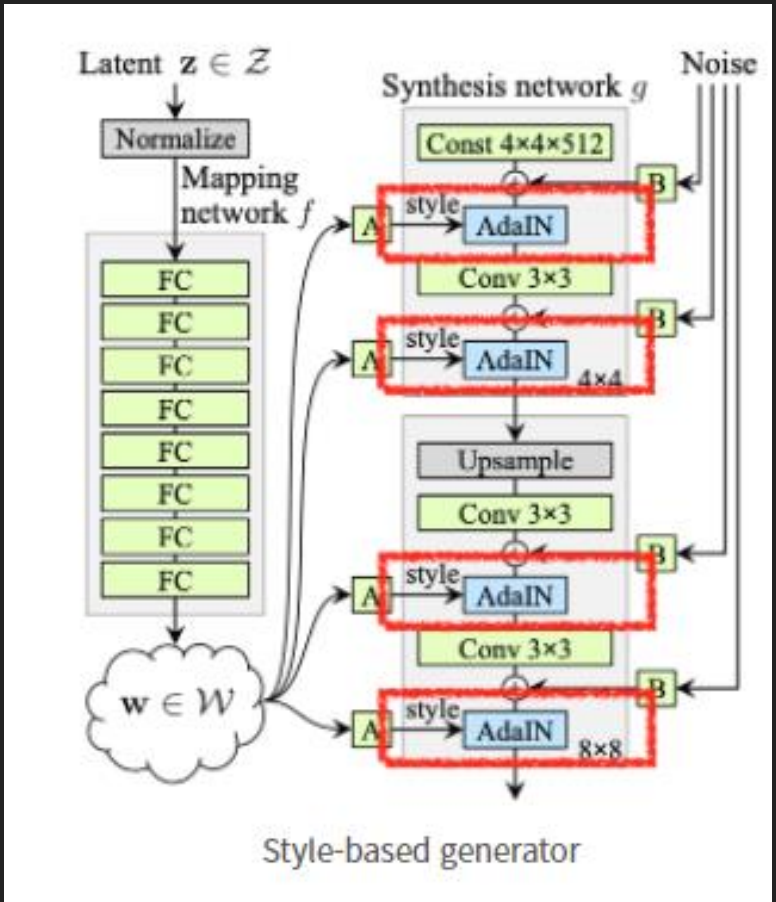
Figure 2: We show examples of cars (first two rows) synthesized in chosen viewpoints (columns). To get these, we fix the latent code w_v^* that controls the viewpoint (one code per column) and randomly sample the remaining dimensions of (Style)GAN's latent code (to get rows). Notice how well aligned the two cars are in each column. In the third row we show the same approach applied to horse and bird StyleGAN.

논문 저자의 첫 번째 목표는 style-gan을 이용하여 여러 시점의 이미지를 만드는 것이다.

Style-gan은 16개의 각각 가중치의 정규 분포로 매핑된 뉴럴 네트워크로 이어져있습니다.

각각의 레이어는 다른 이미지 속성을 제어하는데 레이어의 초기 부분은 카메라의 시점을 제어하는 것을 알아냈고 높은 레이어 부분은 모양과 색상 배경을 제어하는 것을 알아냈습니다.

STYLEGAN AS SYNTHETIC DATA GENERATOR



Style-based generator

16개의 레이어는 StyleGan에서 2개의 block으로 처리가 되어있고, 총 4개의 레이어에서 카메라 시점과 관련있는 것으로 보여 2개의 block에 대한 가중치를 변경하여 객체에 대하여 여러 시점 데이터를 생성해냅니다.

이때 나머지 12개의 레이어에 대해서는 고정하고 진행을 합니다.

STYLEGAN AS SYNTHETIC DATA GENERATOR

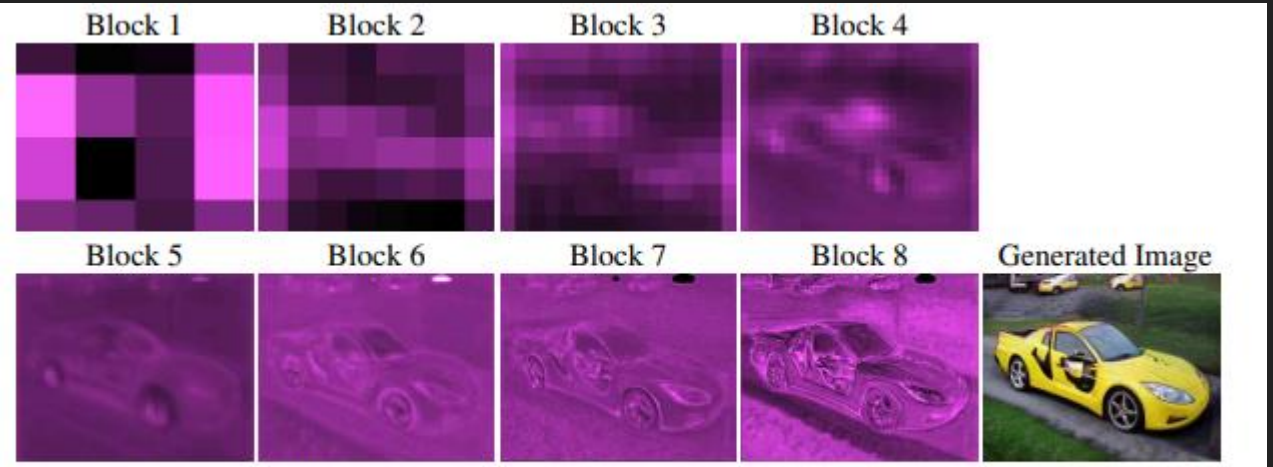


Figure A: **Layer Visualization for Each Block**: Notice that the car contour starts to appear in blocks 4 and higher. This supports some of our findings that the early blocks control viewpoint (and other global properties), while shape, texture and background are controlled in the higher layers.

실제 style-gan의 블록 별 시각화한 자료

블록2까지는 객체의 모양에 대해서 나타나지 않지만, 블록3부터 모양이 잡히는 것을 볼 수 있다.

STYLEGAN AS SYNTHETIC DATA GENERATOR

CAMERA INITIALIZATION

이러한 과정으로 생성된 데이터의 장점이 또 존재하는데 이미지 주석처리가 간단해지는데 그 이유는 시점 처리 w를 제외하고 나머지 viewpoint를 제어하는 layer의 w는 동일하기 때문에 주석처리가 자동으로 금방 이루어진다.

그렇기 때문에 처음 input으로 들어온 이미지 데이터만 주석처리를 따로 해주고 나머지 시점이 바뀐 데이터에 대한 주석처리는 금방 이루어지게 된다. 이는 기존 다른 데이터 셋에는 200-350시간이 소요되는 작업이지만, 이러한 과정을 거친 데이터의 주석처리는 3~4시간 만에 해결이 가능해진다.

단, 단점이 존재한다면 객체에 있어 많은 관절 포인트를 가지고 있는 경우이다. 이는 시점을 변경할 때 객체가 멈춰 있는 상태가 아닌, 약간의 움직임의 변화가 주어진 경우의 이미지로 생기는 경우가 생겨 주석처리에 어려움이 생길 수 있다. 이 논문에서는 어느 정도의 포인트를 가진 말과 새까지 표현하였으나, 더욱 많은 포인트를 가진 객체에 대해서는 한계점을 가진다고 나와있다.

STYLEGAN AS SYNTHETIC DATA GENERATOR

CAMERA INITIALIZATION

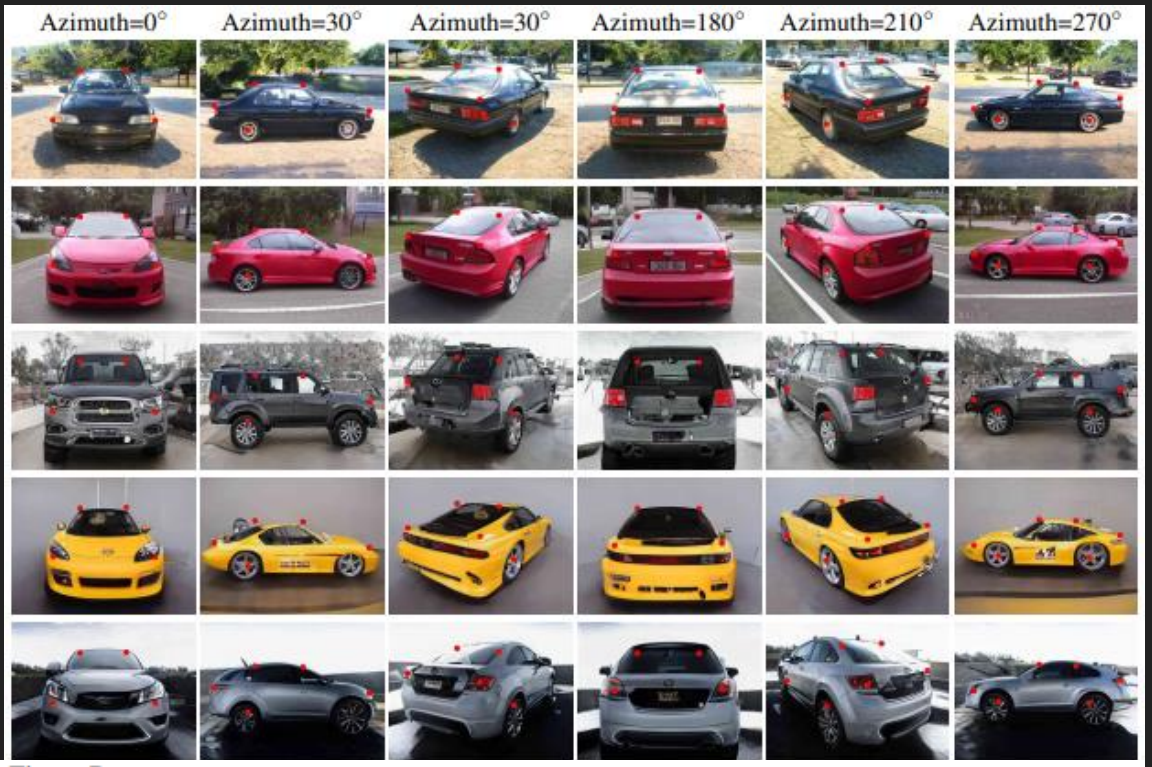
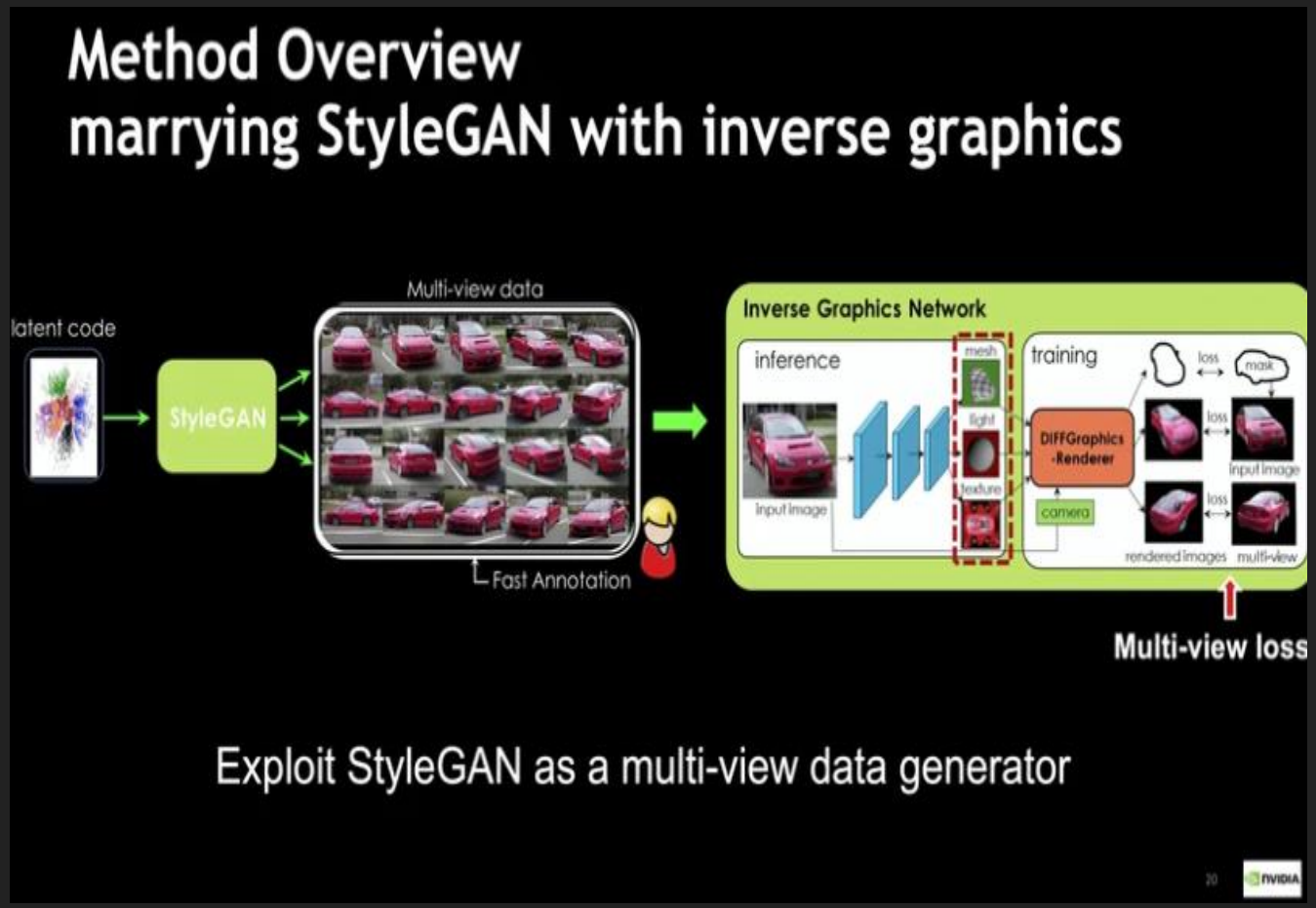


Figure D: We show examples of cars synthesized in chosen viewpoints (columns) along with annotations. Top row shows the pose bin annotation, while the images show the annotated keypoints. We annotated keypoints for the car example in the first image-row based on which we compute the accurate camera parameters using SfM. To showcase how well aligned the objects are for the same viewpoint latent code, we visualize the annotated keypoints on all other synthesized car examples. Note that we do not assume that these keypoints are accurate for these cars (only the implied viewpoint). Annotating pose bins took 1 min for the car class, while keypoint annotation took 3-4 hours, both types of annotations thus being quite efficient. We empirically find that pose bin annotation is sufficient in training accurate inverse graphics networks (when optimizing camera parameters during training in addition to optimizing the network parameters).

Horse Viewpoints

Figure B: **All Viewpoints:** We show an example of a car, bird and a horse synthesized in all of our chosen viewpoints. While shape and texture are not perfectly consistent across views, they are sufficiently accurate to enable training accurate inverse graphics networks in our downstream tasks. Horses and birds are especially challenging due to articulation. One can notice small changes in articulation across viewpoints. Dealing with articulated objects is subject to future work.

TRAINING AN INVERSE GRAPHICS NEURAL NETWORK



위에서 만든 다시점 데이터를 통해 각각의 데이터에서 Inverse Graphics Network를 이용한다. 이는 2D데이터를 3D화 시켜주는 학습 과정으로 2D 데이터에서 3D화에 필요한 속성들을 잡아낸다. (객체의 모양, 빛, 객체의 색상) 그리고 이를 통해 만들어낸 3d화 이미지를 Input 이미지와 비교하여 학습을 진행한다. 학습을 위한 손실 함수는 각각의 이미지의 부분에 대한 함수를 이용한다.

$$L(I,S,T,V;\theta) = \lambda_{\text{col}} L_{\text{col}}(I,I') + \lambda_{\text{percpt}} L_{\text{percpt}}(I,I') + L_{\text{IOU}}(M,M') + \lambda_{\text{sm}} L_{\text{sm}}(S) + \lambda_{\text{lap}} L_{\text{lap}}(S) + \lambda_{\text{mov}} L_{\text{mov}}(S) \quad (1)$$

그리고 다양한 시점이 존재하기에 시점마다의 값을 합하여 Multi-view loss를 줄이기 위해 학습을 진행하게 된다.

$$\mathcal{L}_k(\theta) = \sum_{i,j,i\neq j} \left(L(I_{V_i^k}, S_k, T_k, V_i^k; \theta) + L(I_{V_j^k}, S_k, T_k, V_j^k; \theta) \right)$$

$$\text{where } \{S_k, T_k, L_k\} = f_\theta(I_{V_i^k})$$

DISENTANGLING STYLEGAN WITH THE INVERSE GRAPHICS MODEL

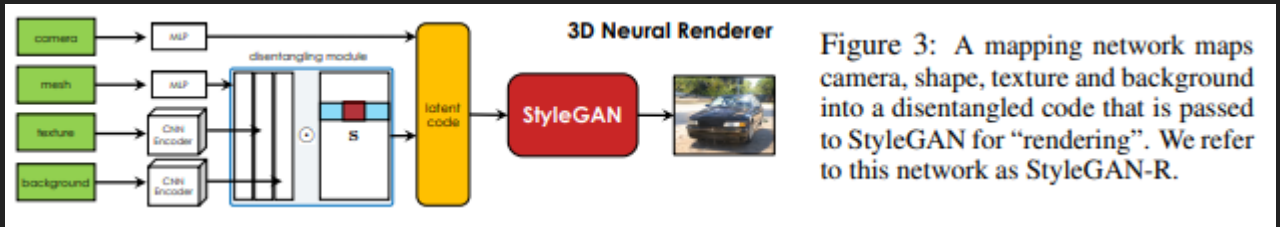
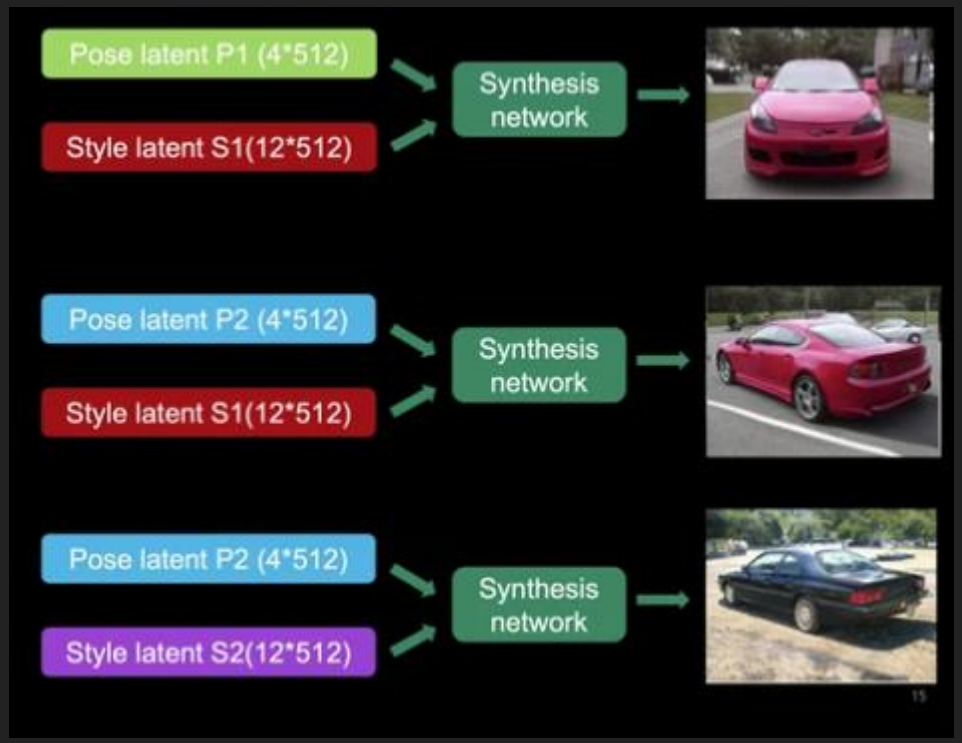


Figure 3: A mapping network maps camera, shape, texture and background into a disentangled code that is passed to StyleGAN for "rendering". We refer to this network as StyleGAN-R.



2d 이미지에서 얻어낸 객체의 속성을 이용하여 다시 StyleGan으로 돌아가 이를 이용할 수 있는데, 이전 StyleGan에서는 시점을 변경하기 위해 사용되었다면 이번에는 객체의 색상, 구조, 배경을 바꾸는데 사용한다.

이전에는 카메라 시점을 변경하기 위해 앞 부분 1~2 block의 가중치를 바꾸고 뒤에 12개의 레이어를 고정하였다면 이번에는 이후 12개의 레이어를 변환하고, 앞 1~2의 block을 고정하면 같은 시점의 다른 모습의 객체 이미지가 생성된다.

그리고 새로 생성된 이미지에 대해서는 다시 여러 시점의 이미지를 생성할 수 있게 되고 3차원 데이터 생성을 위한 학습을 할 수 있게 된다.

DISENTANGLING STYLEGAN WITH THE INVERSE GRAPHICS MODEL

Figure 10: **3D Manipulation:** We sample 3 cars in column 1. We replace the shape of all cars with the shape of Car 1 (red box) in 2nd column. We transfer texture of Car 2 (green box) to other cars (3rd col). In last column, we paste background of Car 3 (cyan box) to the other cars. Examples indicated with boxes are unchanged. Zoom in to see details.

EXPERIMENTS

Datasets

We first randomly sample 6000 cars, 1000 horse and 1000 birds with diverse shapes, textures, and backgrounds from StyleGAN. After filtering out images with bad masks as described in Sec. 3, 55429 cars(39개의 시점으로 진행), 16392 horses(22개의 시점으로 진행), and 7948 birds(8개의 시점으로 진행), images remain in our dataset which is significant larger than the Pascal3D car dataset (Xiang et al., 2014) (4175 car images). Note that nothing prevents us from synthesizing a significantly larger amount of data, but in practice, this amount turned out to be sufficient to train good models.

EXPERIMENTS

Dataset	Size	Annotation
Pascal3D	4K	200-350h
StyleGAN	**50K**	**~1min**

(a) Dataset Comparison

Model	Pascal3D test	StyleGAN test
Pascal3D	**0.80**	0.81
Ours	0.76	**0.95**

(b) 2D IOU Evaluation

	Overall	Shape	Texture
Ours	**57.5%**	**61.6%**	**56.3%**
Pascal3D-model	25.9%	26.4%	32.8%
No Preference	16.6%	11.9%	10.8%

(c) User Study

Table 1: **(a):** We compare dataset size and annotation time of Pascal3D with our StyleGAN dataset. **(b):** We evaluate re-projected 2D IOU score of our StyleGAN-model vs the baseline Pascal3D-model on the two datasets. **(c):** We conduct a user study to judge the quality of 3D estimation.

Conclusion

Limitaion

3D 데이터를 잘 생성해내는 반면(모양, 깊이 등) 그림자 부분에서는 예측이 실패하는 경우가 존재하며, StyleGan을 이용하여 배경 전환에 성공하였으나, 일부분 약간의 이상한 변환이 생기는 경우가 존재했다. 이는 미래 기술에 맡긴다.

Success

StyleGan을 이용하여 3D 모형을 생성해내는데 성공적이였으며 이미지 데이터 주석 처리에 있어 엄청난 효율성을 보였다.

참고 문헌

블로그, 기사

- http://www.aitimes.kr/news/articleView.html?idxno=20817
- https://airsbigdata.tistory.com/217

ICLR 2021

- https://crossminds.ai/video/image-gans-meet-differentiable-rendering-for-inverse-graphics-and-interpretable-3d-neural-rendering-60c3d3776af07cfaf7325f76/

AI4CC Workshop

- https://www.youtube.com/watch?v=DrizewlvZGc&t=1428s

논문

- https://arxiv.org/pdf/2010.09125.pdf

THANK YOU