

The background is a dark blue field filled with a complex network of glowing blue lines and small orange dots, resembling a neural network or a molecular structure. The lines are thin and curved, connecting the dots in a web-like pattern. The dots are small and bright, with some appearing as larger, more intense orange spheres.

Unsupervised Learning of Probably Symmetric Deformable 3D Objects from Images in the Wild

CVPR 2020 best paper

Abstract

- 논문에서 다루고자 하는 task 는 무엇인가요?
 - 외부의 감독(supervision) 없이 하나의 각도로만 찍힌 영상(single view images)로부터 3D deformable object categories 를 학습하는 방식을 제안함.
 - 3D modelling 은 자연 영상(natural image, 2D) 의 변동성의 많은 부분을 설명 할 수 있기 때문에 3D modelling 문제를 deformable(변형 가능한) object categories 로 고려 할 수 있음.
- 논문의 기반이 되는 이론적 근거는 무엇인가요?
 - 논문에서 제안하는 방식은 autoencoder에 기반을 두고 있음.
 - Autoencoder 가 입력 영상을 depth, albedo, viewpoint 와 illumination 으로 분해 (factor) 시킴.

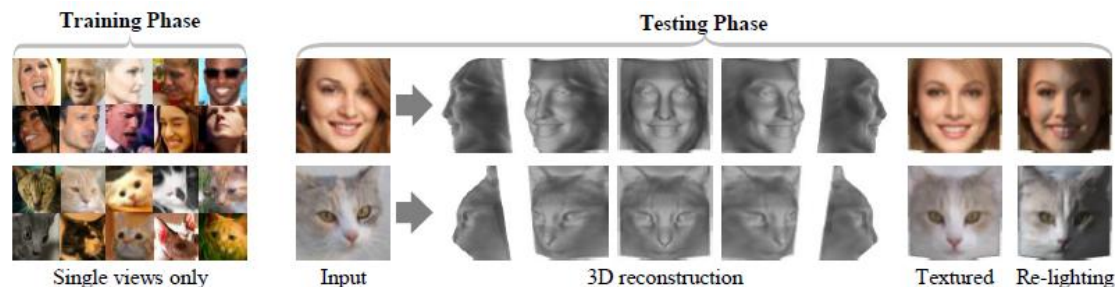


Figure 1: Unsupervised learning of 3D deformable objects from in-the-wild images. Left: Training uses *only* single views of the object category *with no additional supervision at all* (*i.e.* no ground-truth 3D information, multiple views, or any prior model of the object). Right: Once trained, our model reconstructs the 3D pose, shape, albedo and illumination of a deformable object instance from a single image with excellent fidelity. Code and demo at <https://github.com/elliottwu/unsup3d>.

Abstract

- 논문의 핵심 아이디어는 무엇인가요?
 - 어떠한 감독(supervision) 없이 이러한 요소들을 분해하기 위하여 **많은 사물들이 자연적으로 가지고 있는 성질인 물체들이 대칭적인 구조를 이루고 있다는 사실에 착안함.** (가설)
 - *the fact that many object categories have, at least in principle, a symmetric structure.*
 - 논문에서 주장한 가설에 대한 효과는 다음과 같음.
- 1. 조명(illumination)에 대한 추론(reasoning)은 저자들이 내제된 물체의 대칭성 연구를 가능하게 만들었음. (가설 입증)
 - 이는 그림자(shading)으로 인하여 모습이 대칭적이지 않아도 성립됨.
 - Why? 그림자로 인하여 대칭적이지 않게 보이는 것일 뿐이지 **물체의 내제된 속성 자체가 대칭적인 구조를 이루고 있기 때문에** 가능하다고 이해함.
 - *We show that reasoning about illumination allows us to exploit the underlying object symmetry even if the appearance is not symmetric due to shading.*
- 2. 대칭적인 확률 맵 예측을 통하여 물체를 대칭성에 대하여 결정적인(certainly) 방식이 아닌 **확률적인(probably) 방식으로** 모델링하여 모델이 다른 요소(depth, albedo, viewpoint, illumination)을 end-to-end로 예측할 수 있도록 함,
 - Why? *VAEs are directed probabilistic graphical models (DPGM) whose posterior is approximated by a neural network, forming an autoencoder-like architecture.* -> VAE가 확률적으로 예측하는 모델이기 때문이라고 이해함.

Introduction

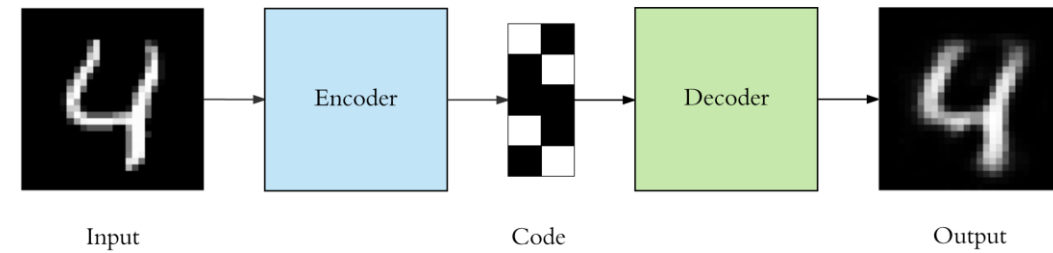
- 저자들은 다음 2개의 도전해야 할 조건(challenge condition)을 가정하여 문제를 정의하고 있음.
 1. 어떠한 2D 와 3D GT 사전정보(key points, segmentation, depth map, ...)를 활용할 수 없음.(unsupervised)
 - 외부의 supervision 없이 학습하는 것이 영상의 annotation 정보를 수집 할 때 발생하는 **bottleneck 현상**을 제거 할 수 있기 때문으로 이 bottleneck 현상은 새로운 application 에서 deep learning 모델을 적용하는 데 있어서 주된 방해 요소가 됨.
 - 새로운 application 에 적용하여 학습하려면 그에 해당하는 annotation 정보도 필요하기 때문이라고 이해.
 2. 알고리즘은 반드시 하나의 각도로만 찍힌 영상(single view images) 에 대하여 제약이 없이 수집을 사용해야함.
 - 특히, 같은 대상에 대하여 다양한 각도로 찍힌 영상(multiple view of the same instance) 을 요구해서는 안됨.
 - 왜냐하면, 많은 application 에서 (특히, 변형가능한 물체에 대하여) 주어진 정보는 단일 영상 하나뿐이기 때문에 하나의 각도로만 찍힌 영상(single view images) 을 가정해야 다양한 실용적인 문제에 적용 가능함.

* Bottleneck 현상 : 병목(bottleneck) 현상은 전체 시스템의 성능이나 용량이 **하나의 구성 요소로 인해 제한을 받는 현상**을 말한다. 병목"이라는 용어는 물이 병 밖으로 빠져나갈 때 병의 몸통보다 병의 목부분의 내부 지름이 좁아서 물이 상대적으로 천천히 쏟아지는 것에 비유한 것이다.

Introduction

- 이렇게 가정된 문제를 해결하기 위하여 autoencoder 는 데이터에 대한 표현을 배우는 모델이기 때문에(확률적 모델) autoencoder를 활용하여 영상을 factors(albedo, depth, illumination, viewpoint) 로 감독 없이(supervision) 분해 할 수 있다는 점에 이론적인 기반을 두고 있음.
- 하지만, 추가적인 가정 없이는 4개의 factors 로 영상을 분해하는 것은 매우 ill-posed problem 임.
 - 저자들은 이에 대한 해결책으로 이러한 ill-posed 문제의 범위를 좁혀주는 최소한의 가정을 찾게 되었고 대부분의 물체들이 대칭적인 속성을 지니고 있다는 점에 주목하였음.

Autoencoder



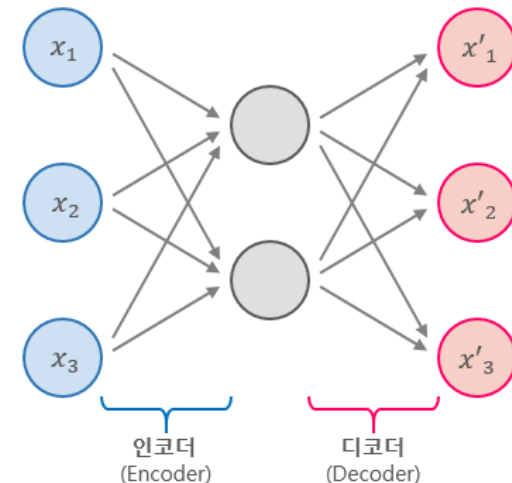
- 오토인코더(Autoencoder)는 unsupervised 방식으로 효율적인 데이터 코딩을 학습하는 데 사용되는 인공 신경망의 일종임.
 - 보통, 오토인코더를 unsupervised learning 기법 이라고 하지만 정확하게 말하면 명확하게 말하면 self-supervised 기법의 일종임. 왜냐하면 학습 데이터로부터 자신의 label를 만들기 때문임.
- 오토인코더(Autoencoder)의 목표는 신호의 잡음(noise)을 무시하도록 네트워크를 훈련시킴으로써 일반적인 차원 축소를 위한 일련의 데이터에 대한 표현을 배우는 것임.
- 입력과 출력이 같은 feedforward neural network의 일종으로 더 낮은 차원으로 (lower-dimensional code)로 압축하는 것을 representation으로 compression 하는 과정과 representation으로 부터 출력물을 reconstruction 하는 과정으로 이루어져 있음.
- 이렇게 입력으로 부터 compression 된 representation 을 code, latent-space representation 이라고 부름.

https://en.wikipedia.org/wiki/Autoencoder#cite_note-:3-32

<https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798>

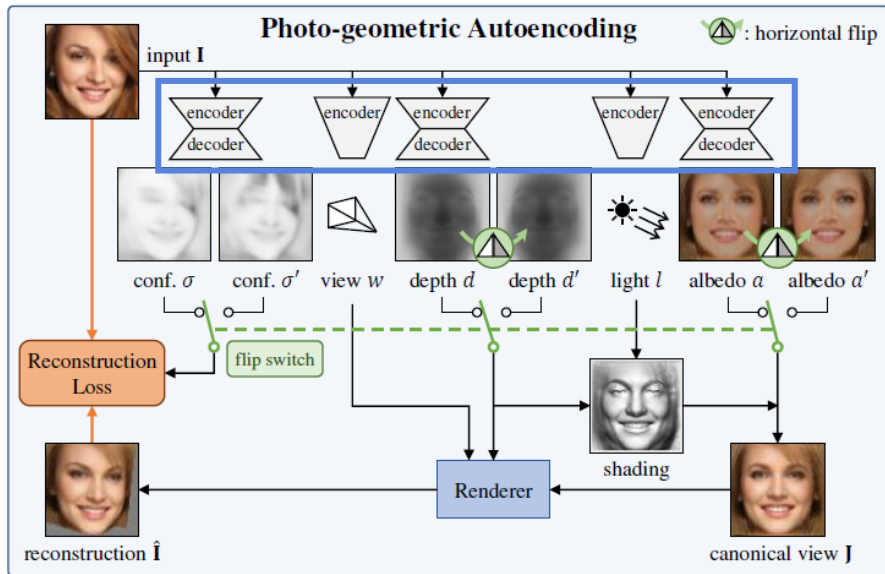
Autoencoder – 개념

- 오토인코더(Autoencoder) 란 무엇인가?
 - 데이터를 효율적으로 coding(여기서, coding 이란 데이터를 압축하는 것을 의미함) 하기 위한 딥러닝 구조의 일종임.
 - 즉, 데이터를 효율적으로 나타내기 위하여 고차원을 저차원으로 차원 축소하는 방법.
 - 입력과 출력이 같은 신경망으로 간단해 보이지만 여러가지 방식으로 제약을 걸어 줌으로써 이 제약들은 단순히 입력을 바로 출력하지 못하도록 함으로써 데이터를 효율적으로 표현(representation) 하는 방법을 학습하도록 함.



Architecture and Code

- 논문에서 제안하는 방식은 autoencoder에 기반을 두고 있음.
- Autoencoder 가 입력 영상을 depth, albedo, viewpoint 와 illumination 으로 분해 (factor) 시킴.



```
## networks and optimizers
self.netD = networks.EDDeconv(cin=3, cout=1, nf=64, zdim=256, activation=None)
self.netA = networks.EDDeconv(cin=3, cout=3, nf=64, zdim=256)
self.netL = networks.Encoder(cin=3, cout=4, nf=32)
self.netV = networks.Encoder(cin=3, cout=6, nf=32)
self.netC = networks.ConfNet(cin=3, cout=2, nf=64, zdim=128)
```

Figure 2: **Photo-geometric autoencoding.** Our network Φ decomposes an input image I into depth, albedo, viewpoint and lighting, together with a pair of confidence maps. It is trained to reconstruct the input without external supervision.

Introduction - 그렇다면, 왜 대칭성이 중요할까?

- 이를 바탕으로 물체가 완벽하게 대칭이라고 가정 한다면, 단순히 영상을 mirroring 하기만 해도 그것의 가상적인 두번째 각도(virtual second view) 을 얻을 수 있음.
 - Why?
- 고로, 만약 mirrored pair 영상 사이의 일치(correspondence) 를 얻을 수 있다면 서로 다른 각도에서 찍힌 2장의 영상에 대한 정보를 알 수 있기 때문에 stereo reconstruction 으로 3D reconstruction 이 가능해짐.
- 이러한 추론에 근거하여, 논문의 저자들은 대칭적인 특성을 geometric cue 로 활용하여 decomposition 에 제약을 걸어 줄 수 있음.

Introduction

- 하지만, 물체는 실제로 완벽히 대칭적이지 않음. (가설의 한계)
 - 예를 들어, shape 측면에서는 얼굴의 포즈 혹은 머리 모양, 표정에 의한 변화로 인하여 대칭적이지 않으며 albedo 측면에서도 고양이의 비대칭적인 texture 와 같이 대칭적이지 않을 수 있음.
- 이러한 문제를 극복하기 위하여 저자들은 두가지 방식으로 문제를 해결 하였음.
 1. 내제된 대칭성(underlying symmetric)에 대하여 연구하기 위하여 **명확하게(explicitly) 조명**에 대하여 모델링함.
 - 이를 통하여, 모델이 조명을 **shape 복원**하는데 있어서 추가적인 단서로 사용 가능함.
 2. 물체의 대칭성의 결여 가능성을 추론하기 위하여 model 을 augment 함.
 - Second, we augment the model to reason about potential lack of symmetry in the objects.
 - 이를 위하여, **다른 factors** 과 함께 **모델은 주어진 픽셀이 영상에서 대칭적인 대응물을 가지고 있을 확률을 포함하는 dense map** 을 예측함.

Model 을
augment 한다는
것이 무슨 말인가?

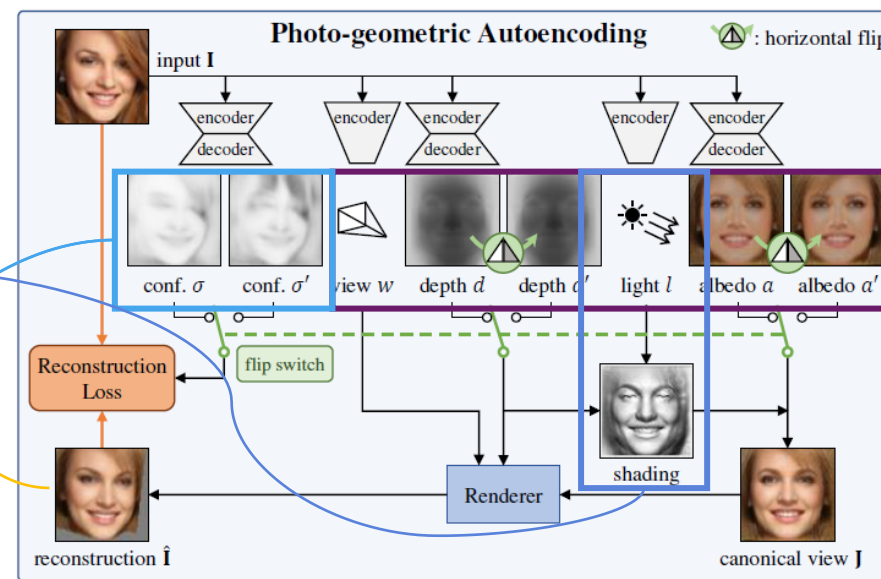
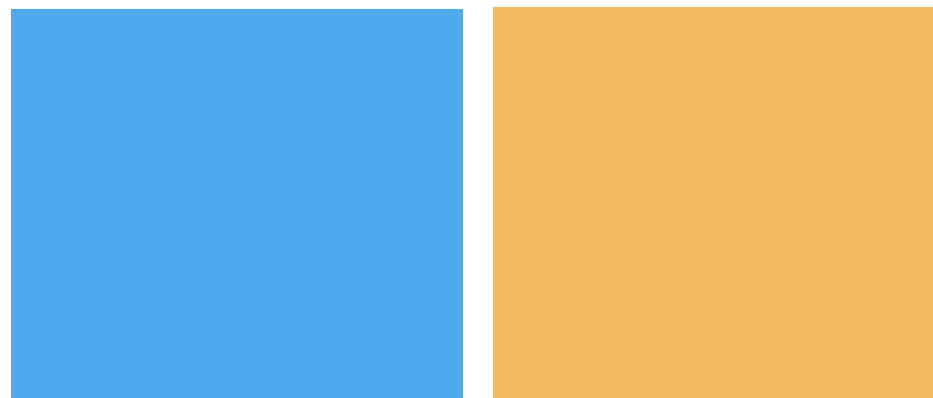


Figure 2: **Photo-geometric autoencoding.** Our network Φ decomposes an input image I into depth, albedo, viewpoint and lighting, together with a pair of confidence maps. It is trained to reconstruct the input without external supervision.

Introduction

- 결국, 이러한 요소들을 end-to-end learning formulation 으로 표현 할 수 있음.
 - 여기서, 말하는 요소들은 raw RGB 영상에서만 학습되는 confidence map 을 포함.
- 게다가, 저자들은 내부적인 표현(internal representation)을 flipping 시킴으로써 대칭성을 강제 될 수 있음을 보여주는데 이는 특히 확률적으로 대칭에 대한 추론이 가능함을 의미하고 있음.
 - We also show that symmetry can be enforced by flipping internal representations, which is particularly useful for reasoning about symmetries probabilistically.

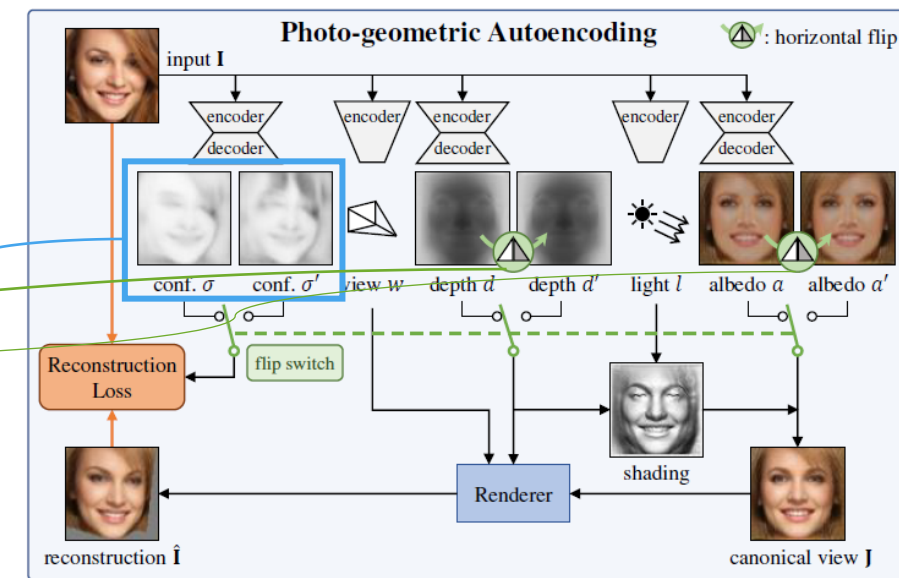


Figure 2: **Photo-geometric autoencoding.** Our network Φ decomposes an input image I into depth, albedo, viewpoint and lighting, together with a pair of confidence maps. It is trained to reconstruct the input without external supervision.

SMPL

- 그래서 인간의 행동을 "자세하게" 또는 "정확히" 이해하기 위해서는 신체의 주요 관절 뿐만 아니라 모든 관절에 대해 인식해야 하며, 전체 관절에 대한 3차원 표면이 필요하다는 것이 이 논문의 주장이다.
- 그래서 논문에서는 인간의 얼굴, 손, 그리고 신체 구성의 복잡한 모든 것들을 표현할 수 있는 3차원 모델이 필요하다는 것이 논문의 주장이었었고, 두번째는 단일 이미지 즉, 이미지 하나로 이러한 모델을 추출하는 것이 필요하다고 했다.
- 얼굴 복원도 마찬가지 아닐까?
 - 모든 것이 필요하다..

Basel

- 얼굴 인식 task 에서 딥러닝 이전 세대에서 많은 주목을 받은 논문임.
 - The resulting model parameters separate pose, lighting, imaging and identity parameters

Method

- 사람의 얼굴과 같은 unconstrained collection of images of an object category 가 주어졌을 때, 목표는 image of object instance(객체 인스턴스의 영상)을 입력 받아 이것을 분해 시켜(decomposition) 3D shape, albedo, illumination, viewpoint 과 같은 factor 로 출력 시켜주는 모델을 학습하는 것임.
- 하지만, 문제의 조건(unsupervised)에 의하여 주어진 정보는 raw RGB 영상밖에 없기 때문에 이를 복원(reconstruction) 하도록 하는 것이 목적 함수임. (reconstruction loss)
 - 즉, 모델은 위 네가지 factors 를 조합하여 입력 영상으로 돌리도록 학습 하도록 설계함.
 - 고로 이를 위하여, autoencoder 를 사용하여 4가지 factors로 분해.

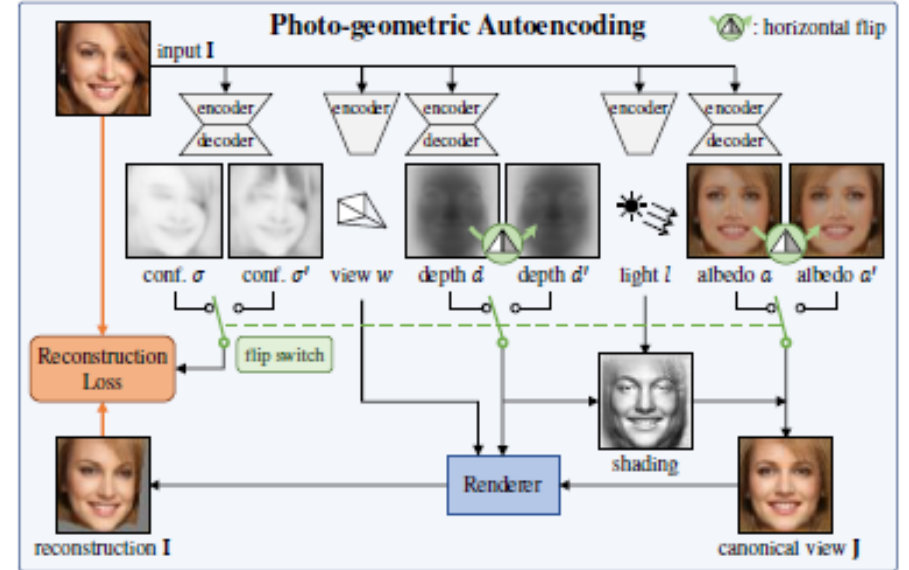
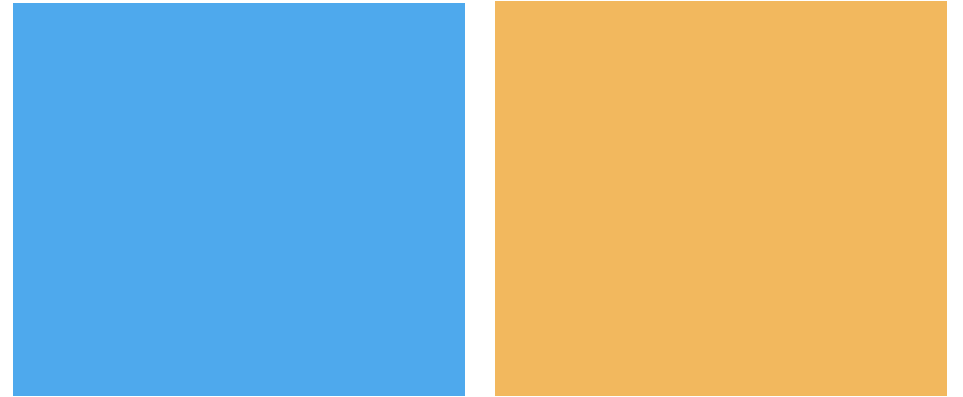


Figure 2: Photo-geometric autoencoding. Our network Φ decomposes an input image I into depth, albedo, viewpoint and lighting, together with a pair of confidence maps. It is trained to reconstruct the input without external supervision.

Method

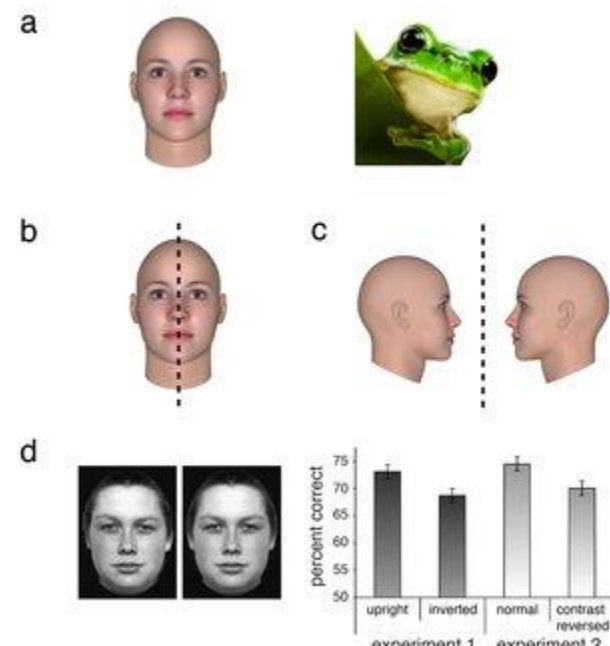
- 논문에서 unsupervised 방식으로 예측이 가능하기 위하여 주목한 점은 대부분의 물체가 대칭적인 특성을 가지고 있다는 점임.
 - *we use the fact that many object categories are **bilaterally symmetric**.*

* Bilaterally symmetric (생물학적인 대칭) : 생물학의 대칭 또는 생물학의 대칭형은 **생물**의 체 내에서 볼 수 있는 대칭 배열 모양이다. 극축이 있으면 이것을 중심으로 한 대칭 관계가 나타나는데, **수학적·물리적인 뜻의 대칭만큼 엄밀한 것은 아니다**. 방사대칭은 식물에서 매우 일반적으로 볼 수 있다. 관속 식물의 줄기나 뿌리의 구조, 특히 관다발의 배열은 그 좋은 예가 된다. 일반적으로 식물체에는 둥근 기둥 모양의 부분이 많아서, 그 내부 구조가 방사대칭인 경우가 많다. 또 꽃받침, 꽃잎 등에서와 같이 꽃의 각 요소가 방사대칭적으로 배열되어 있는 경우도 많다. 반면, 잎의 경우에는 축에 대하여 좌우대칭 형태를 취하고 있는 것이 많다. 그러나 정확한 대칭이 아니고, 다소 대칭성이 흐트러진 잎(뽕나무)도 있으며, 베고니아의 잎처럼 비대칭적인 것도 있다.

https://ko.wikipedia.org/wiki/%EC%83%9D%EB%AC%BC%ED%95%99%EC%9D%98_%E%B%8C%80%EC%B9%AD

하지만, 이는 완벽하게 대칭을 이루는 것이 아님.

▶ 비대칭성은 shape, albedo, asymmetric illumination 과 같은 요인에 의하여 발생 할 수 있음. 이를 해결하기 위하여 비대칭성에 대한 두가지 측정 방법을 제안함.



<face are bilaterally symmetric>

<https://www.researchgate.net/figure/Faces-are-bilaterally-symmetric-a-Human-and-frog->

Method

- 비대칭성을 측정하기 위한 두가지 방법.
 - we explicitly model asymmetric illumination.
 - our model also estimates, for each pixel in the input image, a confidence score that explains the probability of the pixel having a symmetric counterpart(σ, σ') in the image.

1. 내제된 대칭성(underlying symmetric)에 대하여 연구하기 위하여 명확하게(explicitly) 조명에 대하여 모델링함.

이를 통하여, 모델이 조명을 shape 복원하는데 있어서 추가적인 단서로 사용 가능함.

2. 물체의 대칭성의 결여 가능성을 추론하기 위하여 model 을 augment 함.

Second, we augment the model to reason about potential lack of symmetry in the objects.

이를 위하여, 다른 factors 과 함께 모델은 주어진 픽셀이 영상에서 대칭적인 대응물을 가지고 있을 확률을 포함하는 dense map 을 예측함.

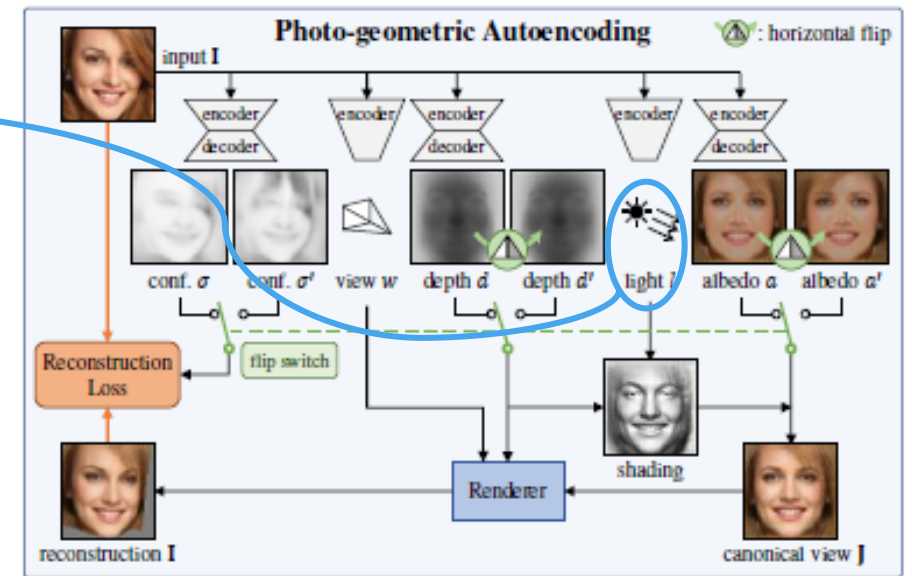


Figure 2: Photo-geometric autoencoding. Our network Φ decomposes an input image I into depth, albedo, viewpoint and lighting, together with a pair of confidence maps. It is trained to reconstruct the input without external supervision.

Method – Photo geometric autoencoding

- I 라는 입력 영상이 주어져 있을 때, 영상이 대략적으로 관심 영역 중심에 위치되어 있다고 가정하자.(symmetric 한 속성을 좀 더 쉽게 활용하기 위함이라고 이해)
 - 목표는 depth map, albedo image, light direction, viewpoint 와 같은 4가지 구성 요소(d, a, w, l)를 예측하는 함수 ϕ 를 neural network 로 모델링 하는 것임.
 - Depth map $d : \Omega \rightarrow \mathbb{R}_+$
 - Albedo image $a : \Omega \rightarrow \mathbb{R}^3$
 - Light direction $l : l \in \mathbb{S}^2$
 - Viewpoint $w : w \in \mathbb{S}^6$
- Ω 는 입력 영상 I 가 정의된 grid 임.
 $\Omega = \{0, \dots, W-1\} \times \{0, \dots, H-1\}$
 (2차원 pixel grid 라고 이해함.)

Depth 를 예측하는 것은 차원이 여러 개 이기 때문임.

여기서 Cost volume이란 두 영상의 강도(Intensity)의 차이를 픽셀 단위로 계산하여 유사도를 측정하는 것을 말함.
 픽셀 단위로 계산된 매칭 비용을 쌓은 것을 Cost volume이라 부름.

각 픽셀 단위로 여러 개를 쌓을 수 밖에 없기 때문에 차원이 증가하게 된다고 이해함.

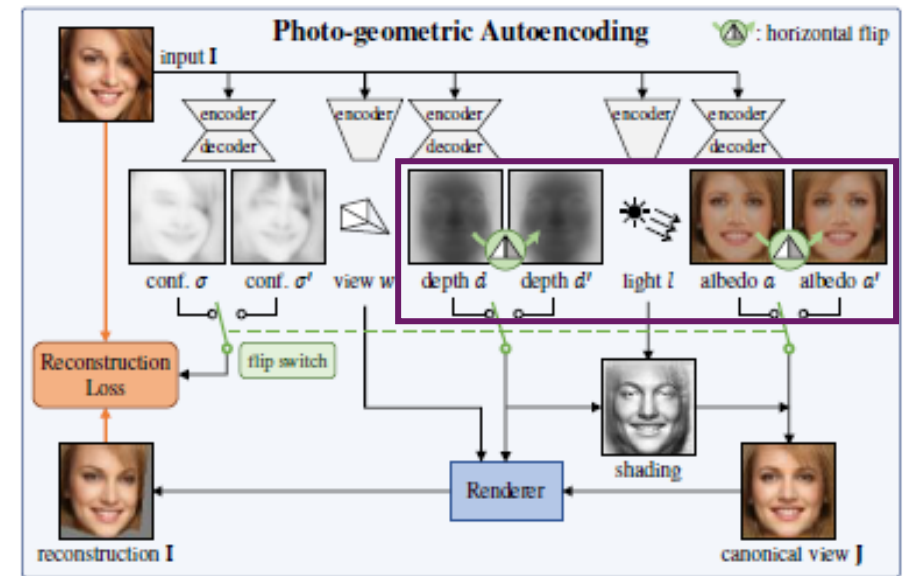
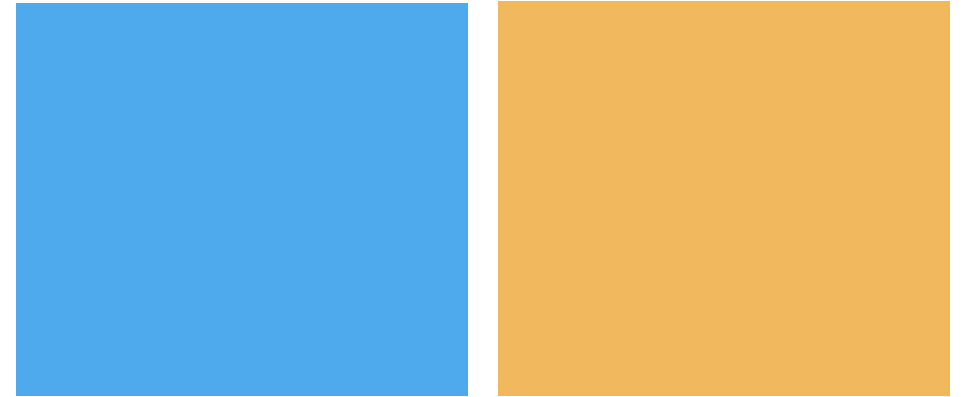
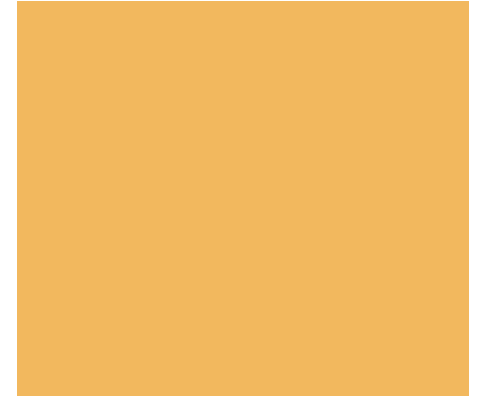


Figure 2: Photo-geometric autoencoding. Our network Φ decomposes an input image I into depth, albedo, viewpoint and lighting, together with a pair of confidence maps. It is trained to reconstruct the input without external supervision.

Method – Photo geometric autoencoding



- 입력 영상 I 가 위 4개의 factor 로 부터 복원되는 과정은 2개의 step 으로 나눌 수 있음. $\Rightarrow \hat{I} = \Pi(\Lambda(a, d, l), d, w)$
 - Lighting Λ
 - Albedo, depth, light (a, d, l) 를 활용하여 view w 의 변화가 개입되지 않은 (canonical view, $w = 0$ 인 정면) canonical image, J 생성함. $\Rightarrow J = \Lambda(a, d, l)$
 - Reprojection Π
 - Viewpoint, w 의 역할은 canonical view 와 실제 입력 영상, I 사이의 transformation 을 기술함.
 - 주어진 shaded canonical image, $J = \Lambda(a, d, l)$ 와 canonical depth, d 를 이용하여 앞에서 설명한 viewpoint, w 를 가지고 복원된 영상을 출력함. 이때, viewpoint 의 역할이 중요한데 reprojection function 은 viewpoint 의 변화의 정보로 인해서 canonical view 에서 복원된 영상, \hat{I} 이 생성됨.
- 이렇게 출력 예측 영상, \hat{I} 를 이용하여 입력 영상 $I \approx \hat{I}$ 이 되도록 학습됨.

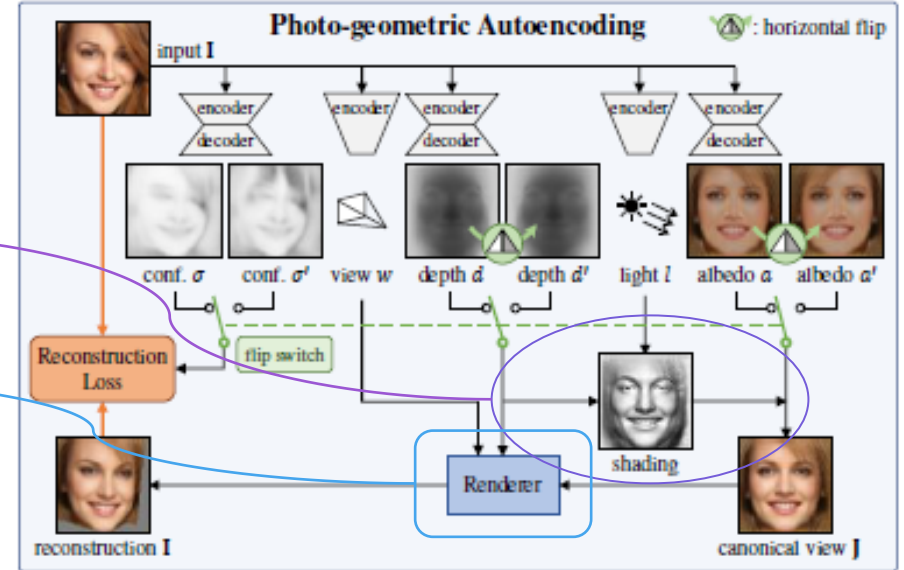


Figure 2: Photo-geometric autoencoding. Our network Φ decomposes an input image I into depth, albedo, viewpoint and lighting, together with a pair of confidence maps. It is trained to reconstruct the input without external supervision.

Method – probably symmetric objects

- 대칭성을 활용해서 3D reconstruction 할 때, 필요한 것은 영상에서 어떠한 points 가 대칭적인지 식별하는 것임.
 - 여기서, 저자들은 canonical frame 으로 복원된 depth, albedo 가 고정된 수직면에 대칭적이라고 가정하였음.
 - Here we do so implicitly, assuming that depth and albedo, which are reconstructed in a canonical frame, are symmetric about a fixed vertical plane.*
 - Why? 이러한 가정이 복원(reconstruction)에 중요한 역할을 하는 ‘canonical view’를 얻어내는 데 핵심적인 역할을 하기 때문임.

이러한 가정에 내제된 저자들의 의도는?
우선 비 대칭적인 속성은 나중에 고려하고 먼저 대칭적이라고 가정하고 canonical view 먼저 쉬운 것 부터 예측해야 훨씬 용이하게 문제를 해결하기 쉽기 때문이라고 추론. 하였지만..
논문에서 나왔듯이 확률적인 대칭에 대한 추론을 위함 임. 그렇다면 왜 확률적인 대칭에 대한 추론이어야 하는가?

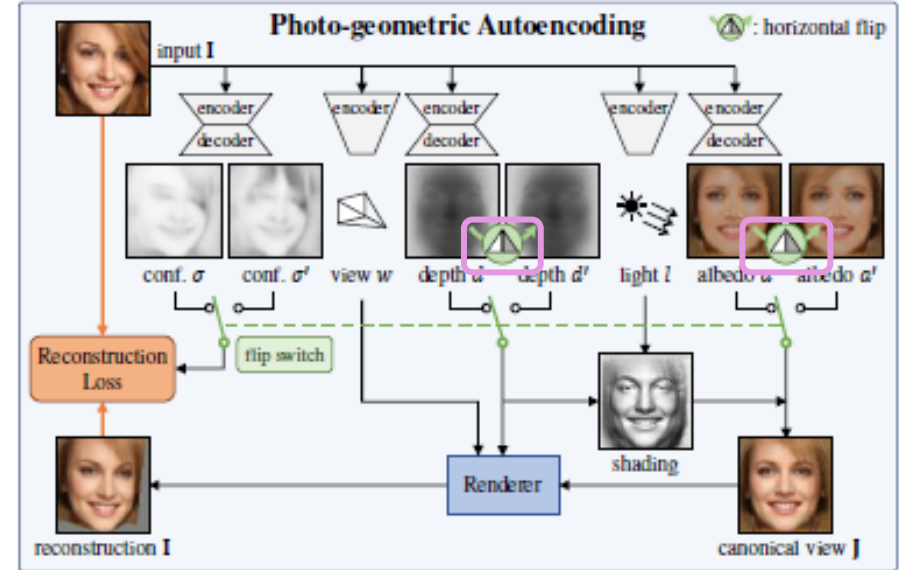


Figure 2: Photo-geometric autoencoding. Our network Φ decomposes an input image I into depth, albedo, viewpoint and lighting, together with a pair of confidence maps. It is trained to reconstruct the input without external supervision.

게다가, 저자들은 내부적인 표현(internal representation)을 flipping 시킴으로써 대칭성을 강제 될 수 있음을 보여주는데 이는 특히 확률적으로 대칭에 대한 추론이 가능함을 의미하고 있음.

Method – probably symmetric objects

- 대칭성을 활용해서 3D reconstruction 할 때, 필요한 것은 영상에서 어떠한 points 가 대칭적인지 식별하는 것임.
 - 여기서, 저자들은 canonical frame 으로 복원된 depth, albedo 가 고정된 수직면에 대칭적이라고 가정하였음.
 - 이러한 가정을 위한 방법은 horizontal axis 에 있음.
 - 출력 맵, $a \in \mathbb{S}^{C \times W \times H}$ 가 주어졌을 때, horizontal axis 를 따라서 반전(flip) 시킴. $\Rightarrow [\text{flip } a]_{c,u,v} = a_{c,W-1-u,v}$
 - 이후, 이러한 가정을 만족시키기 위하여 $a \approx \text{flip } a'$, $b \approx \text{flip } b'$ 가 되도록 강제 시켜 줌.
 - how? 이때, 이에 대응하는 목적함수 corresponding loss 로 강제 시켜 주면 균형(balance)가 안 맞을 수 있기 때문에 (균형이 안 맞는다는 것의 의미를 auxiliary loss 를 reconstruction loss 와 선형 결합해서 최적화 시키기 어렵다는 것으로 이해함) 대신에, 같은 효과를 누리기 위하여 간접적인 방식을 취함. 간접적인 방식이란 을 flipped 된 depth 와 albedo 로 부터 second reconstruction image, \hat{I} 을 얻어내는 방식으로 이루어짐. $\Rightarrow \hat{I}' = \Pi(\Lambda(a', d', l), d', w)$

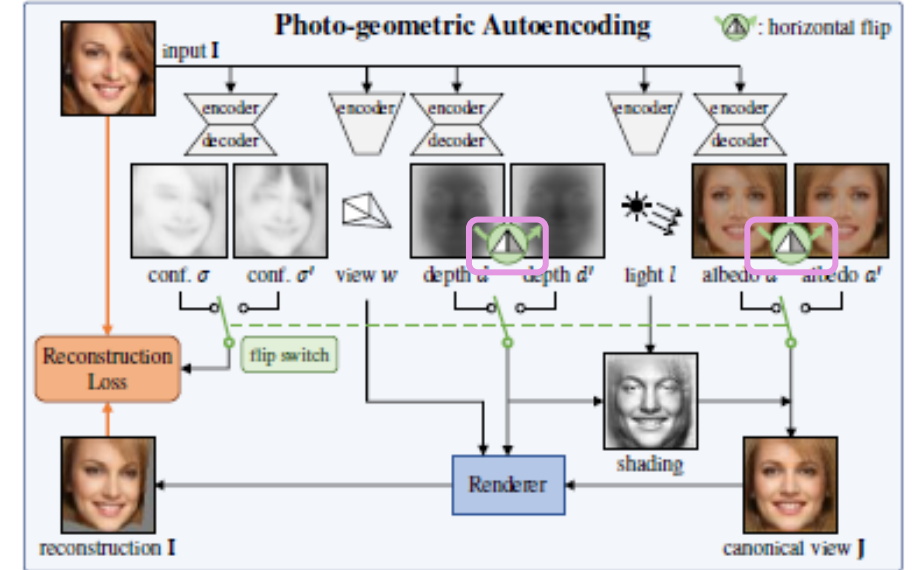
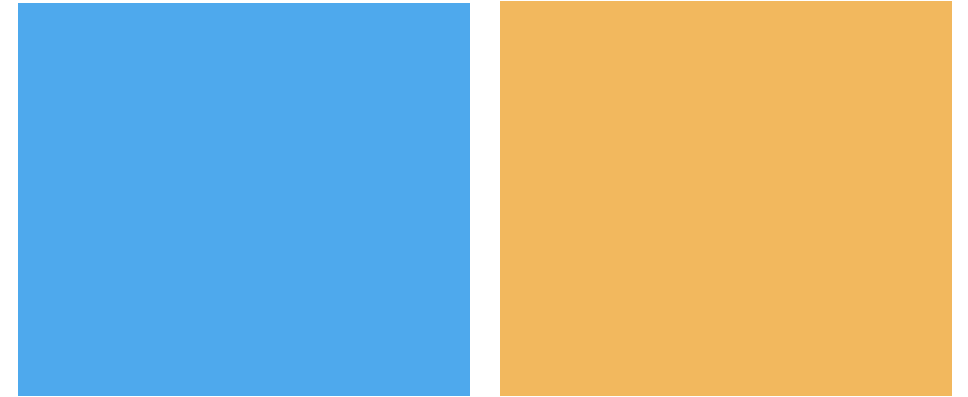


Figure 2: Photo-geometric autoencoding. Our network Φ decomposes an input image I into depth, albedo, viewpoint and lighting, together with a pair of confidence maps. It is trained to reconstruct the input without external supervision.

Method – probably symmetric objects

- 그렇다면, 이것이 직접적인 방식으로 loss 를 주는 것 보다 효과적인 이
유는 무엇일까?
 1. 두 목적함수가 상응(commensurate) 하기 때문에 학습 시, 균형을 맞추기
쉬움.
 2. 가장 중요한 것은 이러한 접근방식으로 인하여 대칭적인 확률을 추론하기
쉬움.

게다가, 저자들은 내부적인 표현(internal representation)을
flipping 시킴으로써 대칭성을 강제 될 수 있음을 보여주는데 이는
특히 확률적으로 대칭에 대한 추론이 가능함을 의미하고 있음.

Method – probably symmetric objects loss function

- 정답(입력, unsupervised) 영상, 예측 영상, confidence map 이 주어졌을 때, 목적 함수는 아래와 같음.
- $\mathcal{L}(\hat{I}, I, \sigma) = -\frac{1}{|\Omega|} \log \frac{1}{\sqrt{2}\sigma} e^{-\frac{\sqrt{2}|\hat{I}-I|}{\sigma}}$
 - Confidence 가 Laplacian distribution 의 분산으로 대응됨.
 - Why? Confidence↓→probability↓

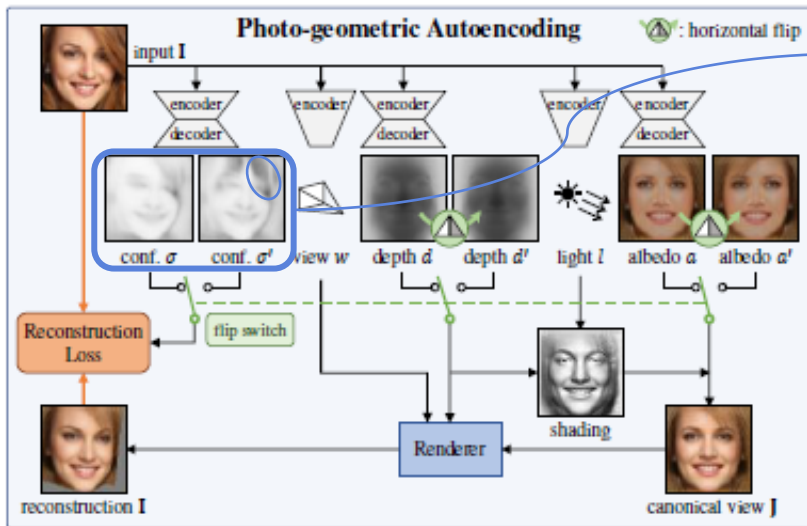
: negative log-likelihood of a factorized Laplacian distribution(L_1) on reconstruction residuals.
- 위와 같은 방식으로 Likelihood(data-term) 을 최적화 해 나가면 confidence map 을 self-calibrate 하는 효과를 이끌어 낼 수 있음.
- 모델의 손실함수는 다음과 같음.
 - $\varepsilon(\phi; \sigma) = \mathcal{L}(\hat{I}, I, \sigma) + \lambda_f \mathcal{L}(\hat{I}', I', \sigma)$
 - 여기서, $\lambda_f=0.5$ 로 설정.
- 그렇다면 왜 Laplacian distribution 을 가정할까?

Maximum Likelihood(관측된 데이터가 나올 확률) estimation 은 데이터가 나올 확률이 최대가 되도록 파라미터를 찾는 방법이다.

https://hyeongminlee.github.io/post/bnn002_mle_map/

Method – probably symmetric objects modeling uncertainty

- 불확실성을 모델링하는 것은 “symmetric” reconstruction (“대칭적” 복원 물, second reconstruction), \hat{I}' 을 고려할 때, 중요성이 더 커짐.
 - 이는 **같은 입력 영상**으로 2차 confidence map (원래의 confidence map, σ 를 horizontal flip) , σ' 을 예측한다는 점이 매우 결정적이기 때문임.
 - 즉, 이로 인하여 입력 영상의 어떤 부분이 대칭적인지 아닌 것인지, 다시 말해, asymmetric 한 불확실성을 예측을 가능하도록 함.



예를 들어, 인간의 얼굴에서 머리와 같은 영역은 비대칭성이 심하게 나타나는 영역이기 때문에 강하게 반응함.

비대칭성을 예측할 때 주의해야 할 점은 대칭성은 *specific instance*(특정 객체) 마다 변화가 심하기 때문에 모델 자체에서 학습되어야 함.

Note that this depends on the specific instance under consideration, and is learned by the model itself.

Figure 2: **Photo-geometric autoencoding.** Our network Φ decomposes an input image I into depth, albedo, viewpoint and lighting, together with a pair of confidence maps. It is trained to reconstruct the input without external supervision.

Method – Image formulation model

Reprojection function: depth map

field of view (FOV) θ_{FOV} 를 가진 perspective camera 를 가정함.

- 주어진 영상은 3차원 물체를 촬영한 카메라에 의하여 얻어진 영상임. 만약, $P = (P_x, P_y, P_z) \in \mathbb{R}^3$ 로 표현된 3차원 공간상에 점(3D point)들은 다음 투영 방식에 의하여 픽셀, $p = (u, v, 1)$ 선형적으로 projection 할 수 있음.

$$\bullet \quad p \propto KP, K = \begin{bmatrix} f & 0 & c_u \\ 0 & f & c_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix}, \begin{cases} c_u = \frac{W-1}{2} \\ c_v = \frac{H-1}{2} \\ f = \frac{W-1}{2 \tan \frac{\theta_{FOV}}{2}} \end{cases}$$

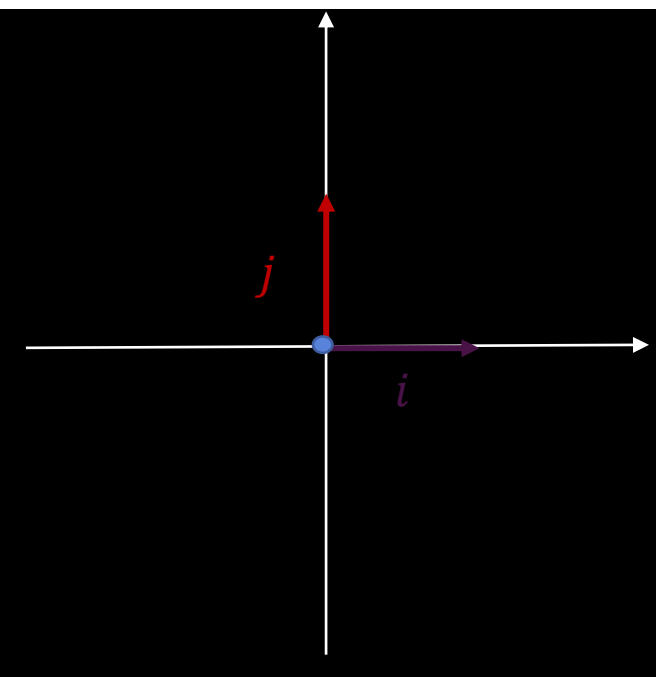
- 네트워크의 출력물
- 이때, 카메라가 물체로부터 떨어진 거리는 약 1m 로 nominal(아주 적은) distance 를 가정함.
 - 특정 물체 주위에 crop 영상이 주어질 때, 상대적으로 적은 FOV 인 10° 를 가정함.

- Depth map 은 앞에서 정의한 camera model, $p \propto KP$ 을 invert 하여 얻어 낼 수 있음. $\Rightarrow P = d_{u,v} K^{-1} p$ 여기서, $d_{u,v}$ 는 depth value 로 depth map 은 depth value, $d_{u,v}$ 에 각 픽셀, $(u, v) \in \Omega$ 을 연관 시킬 수 있음.

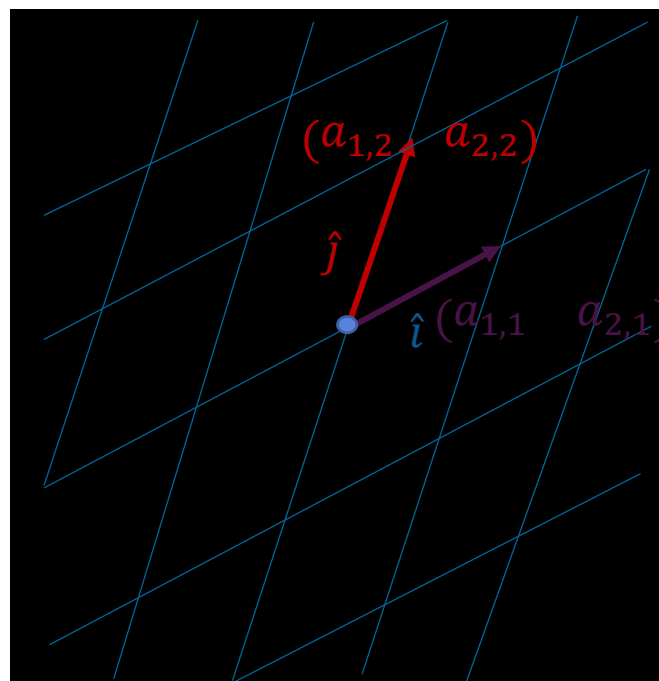
- If we denote with $P = (P_x, P_y, P_z) \in \mathbb{R}^3$ a 3D point expressed in the reference frame of the camera, this is mapped to pixel $p = (u, v, 1)$ by the following projection:

어파인 변환의 동작 원리

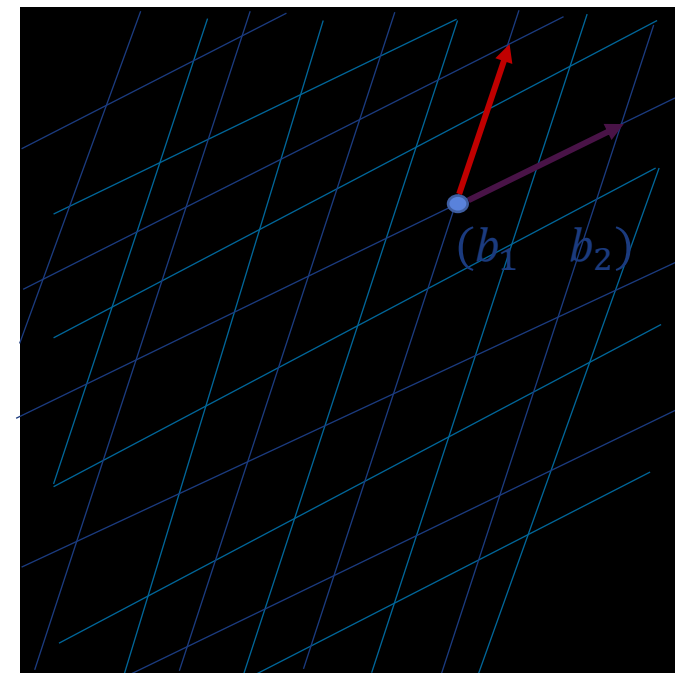
- 어파인 변환 행렬 A b
- $a = \begin{pmatrix} a_{1,1} & a_{1,2} & b_1 \\ a_{2,1} & a_{2,2} & b_2 \\ 0 & 0 & 1 \end{pmatrix}$



\xrightarrow{A}
translation
을 제외한
선형 변환



\xrightarrow{b}
translation



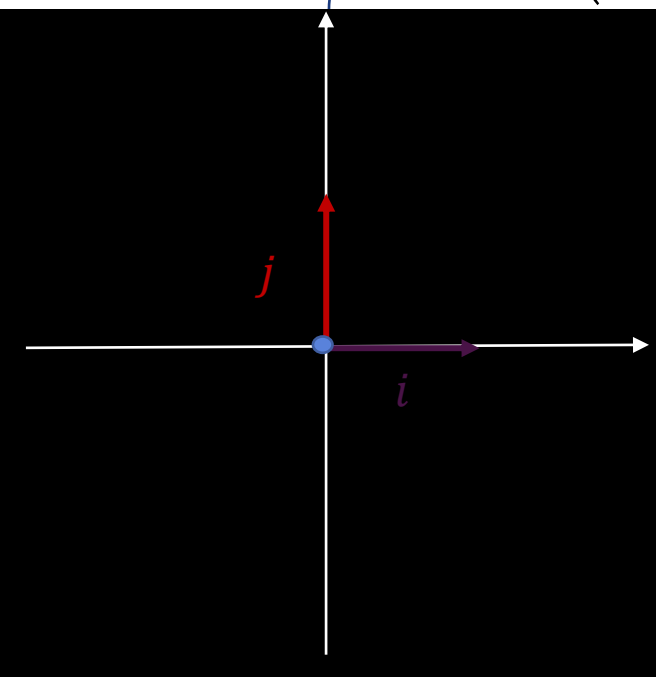
어파인 변환의 동작 원리


- **어파인 변화 행렬**

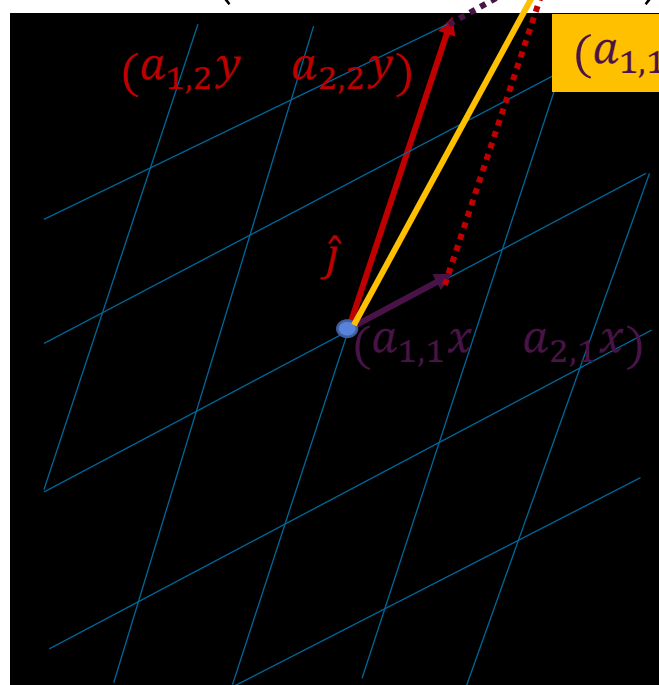
1. translation 이동작

$$\bullet \quad ax = \begin{pmatrix} a_{1,1} & a_{1,2} & b_1 \\ a_{2,1} & a_{2,2} & b_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} a_{1,1}x + a_{1,2}y + b_1 \\ a_{2,1}x + a_{2,2}y + b_2 \\ 1 \end{pmatrix}$$

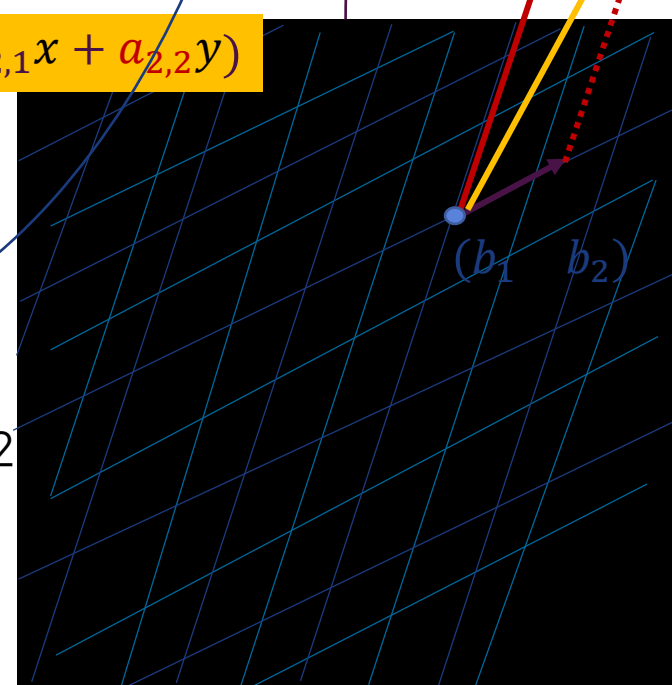
e.g., $x=0.5$, $y=2$



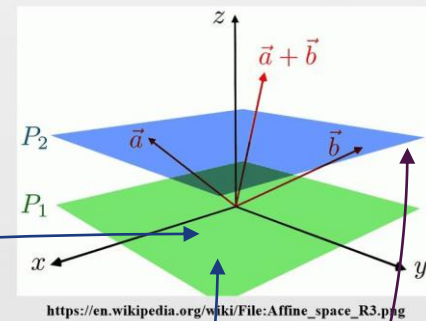

A
translation
을 제외한
선형 변환



Translation
공간1 -> 공간2

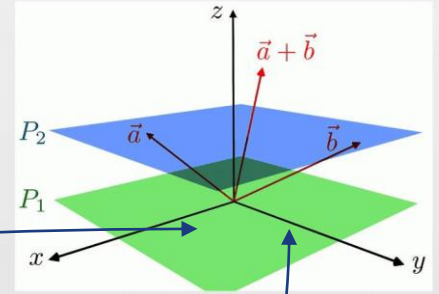


Affine space



공간2
공간1

$$(a_{1,1}x + a_{1,2}y + b_1 \mid a_{2,1}x + a_{2,2}y + b_2)$$



https://en.wikipedia.org/wiki/File:Affine_space_R3.png

공간2
공간1

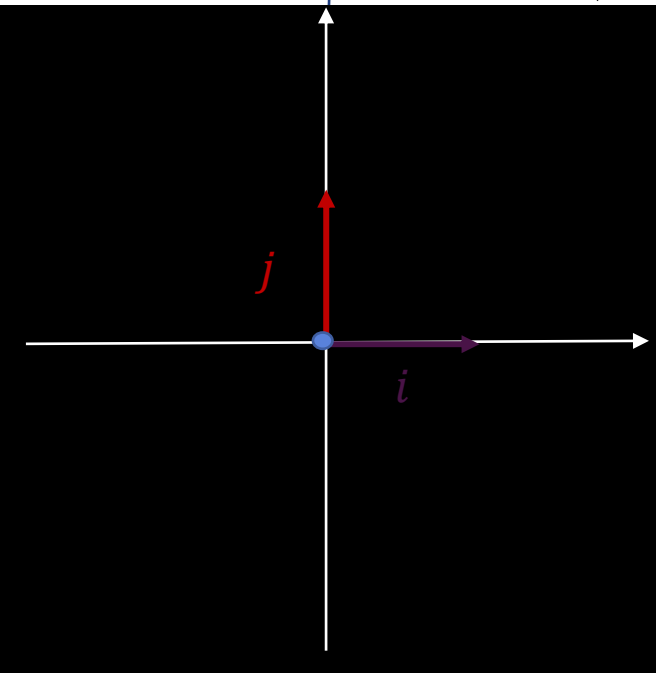
어파인 변환의 동작 원리

- 어파인 변환 행렬

- translation 이 취소

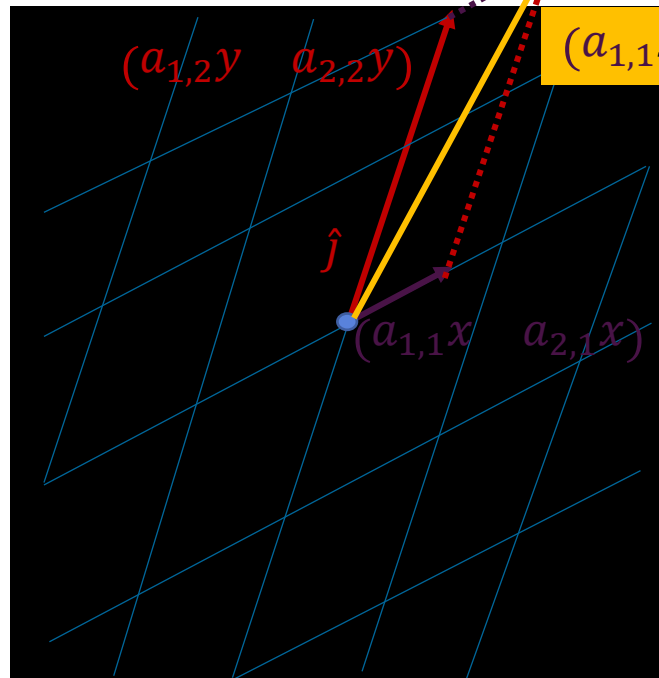
$$ax = \begin{pmatrix} a_{1,1} & a_{1,2} & b_1 \\ a_{2,1} & a_{2,2} & b_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 0 \end{pmatrix} = \begin{pmatrix} a_{1,1}x + a_{1,2}y \\ a_{2,1}x + a_{2,2}y \\ 0 \end{pmatrix}$$

e.g., $x=0.5, y=2$



A

translation
을 제외한
선형 변환



$(a_{1,1}x + a_{1,2}y \quad a_{2,1}x + a_{2,2}y)$

Method – Image formulation model

Reprojection function: viewpoint

- Viewpoint, $w \in \mathbb{S}^6$ 는 Euclidean transformation $(R, T) \in SE(3)$ 을 나타냄.
 - 여기서, $w_{1:3}$ 과 $w_{4:6}$ 은 각 x, y, z 좌표에 대하여 rotation angle과 translations 임.
- 모델에서 viewpoint 가 하는 역할은 canonical view(정면만 가정)에서 actual view (다양한 각도 표현 가능)
 - 고로, canonical view 에서 (u, v) 위치에서의 픽셀은 actual view 에서 (u', v') 에 대응 될 수 있음.
 - 이는 아래와 같은 warping function, $\eta_{d,w}: (u, v) \rightarrow (u', v')$ 으로 표현됨.
 - $p' \propto K(d_{uv} \cdot RK^{-1}p + T)$
 - 여기서, $p = (u, v, 1)$ 이고 $p' = (u', v', 1)$ 에 해당됨.

Method – Image formulation model

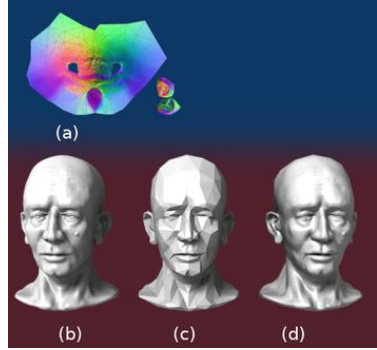
Reprojection function

- 앞에서 depth, d 와 viewpoint change, w 를 입력 받았으면, Reprojection function, $\hat{I} = \Pi(\Lambda(a, d, l), d, w)$ 은 canonical image, $J = \Lambda(a, d, l)$ 에서 다양한 viewpoint 를 가질 수 있는 출력 영상, \hat{I} 을 생성함.
 - How? depth, d 와 viewpoint change, w 를 이용해서 warping 시키는 방식으로 canonical image point, (u, v) 는 actual image point, (u', v') 로 대응 될 수 있기 때문에 warping 시킬 수 있으며 $(u, v) = \eta_{d,w}^{-1}(u', v')$ 로 수식화 할 수 있음.

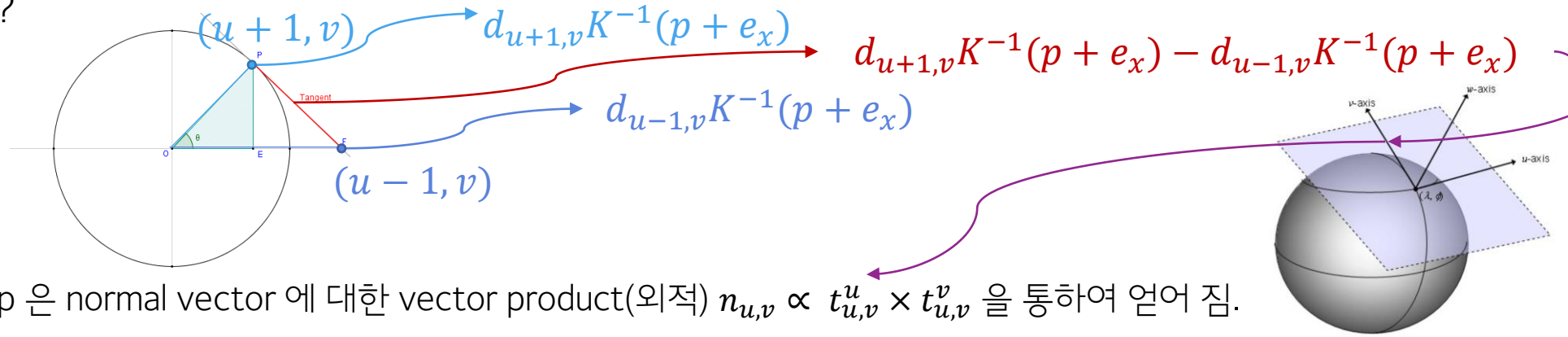
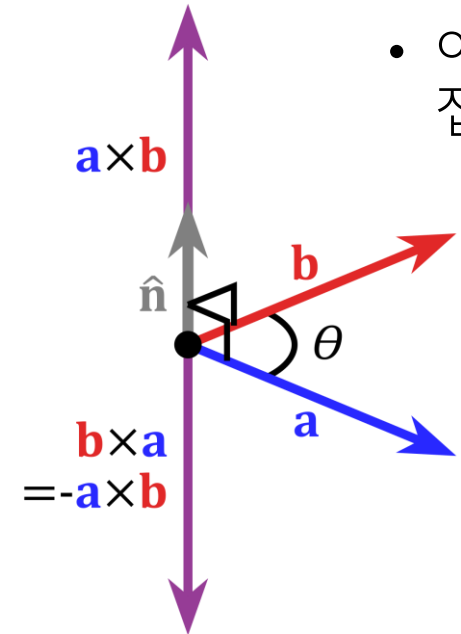
Method – Image formulation model

canonical function: depth map 적용 방법

<Normal mapping used to re-detail simplified meshes. This normal map is encoded in object space>



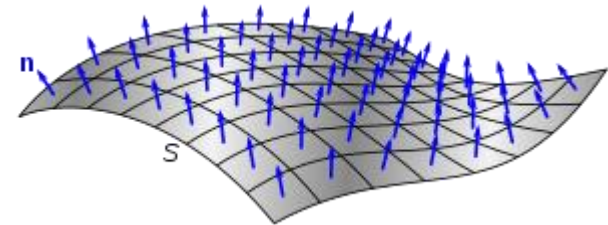
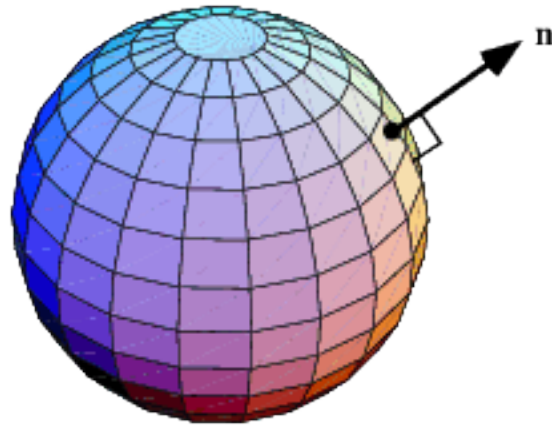
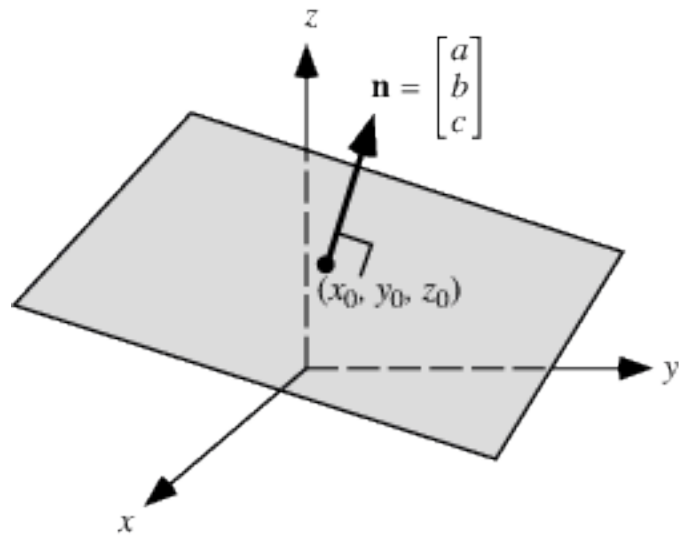
- 그렇다면, albedo, a , depth map, d , light direction, l 이 주어질 때, canonical map, $J = \Lambda(a, d, l)$ 은 어떻게 생성 할 수 있을까?
- Depth map 을 가지고 normal map, $n : \Omega \rightarrow \mathbb{S}^2$ 을 얻어 낼 수 있음.
 - How? 각 픽셀위치 (u, v) 에 대응되는 normal vector 을 내제된 3차원 표면에 연관시켜 normal map 을 얻어 냄.
 - 이때, normal vector 를 얻기 위하여 x-axis 인 u 와 y-axis 인 v 방향으로 3차원 표면에 접하는 (tangent) $t_{u,v}^u, t_{u,v}^v$ 를 계산.
 - 예를 들어서 the first one 은, $t_{u,v}^u = d_{u+1,v}K^{-1}(p + e_x) - d_{u-1,v}K^{-1}(p + e_x)$ 으로 얻어 짐.
 - 여기서, $e_x = (1, 0, 0)$ 으로 x-axis 의 기저 벡터임.
 - Why?



<Vector product> • normal map 은 normal vector 에 대한 vector product(외적) $n_{u,v} \propto t_{u,v}^u \times t_{u,v}^v$ 을 통하여 얻어 짐.

Normal vector 가 3D reconstruction 을 하는데 중요한 이유

- Normal vector 로 3D 로 모델링이 가능하기 때문임.

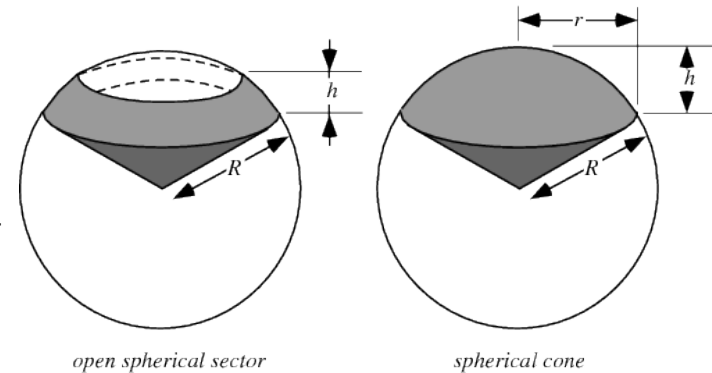


Method – Image formulation model

canonical view: light direction 과 albedo 적용 방법

- light direction (l)
 - 이렇게 얻어진 normal map 을 가지고 light 의 방향, l 을 곱하면 조명의 방향(illumination direction) 을 얻을 수 있음.
빛의 방향 vector Normal vector
- Albedo(빛이 반사하는 정도) (a_{uv})
 - 마침내, 결과는 **illuminated texture** 을 얻기 위하여 다음과 같이 albedo 를 곱함.
 - $J_{uv} = (k_s + k_d \max\{0, \langle l, n_{uv} \rangle\}) \cdot a_{uv}$
 - 여기서, k_s 는 ambient 를 가중치로 가지는 scalar coefficient 이며 k_d 는 diffuse term 임.
 - Why?
 - k_s 와 k_d ambient 와 diffusion 에 대한 weight term 으로 모델이 예측하며 tanh 에 의하여 $[0,1]$ 사이의 값의 범위를 가짐.
 - Why?
 - tanh 를 가지고 통해서 l_x 과 l_y 예측을 통해서 빛의 방향, $l = \frac{(l_x, l_y, 1)^T}{(l_x^2, l_y^2, 1^2)^{0.5}}$ 으로 모델링 되어 spherical sector 로 모델링 될 수 있음.
 - l_x 과 l_y 이 전부 1 안의 범위에 들어오게 하여 point 들을 구면 안에 위치 시킴.

A spherical sector is a [solid of revolution](#) enclosed by two radii from the center of a [sphere](#). The spherical sector may either be "open" and have a conical [hole](#) (left figure; Beyer 1987), or may be a "closed" [spherical cone](#) (right figure; Harris and Stocker 1998).



구면 섹터는 구의 중심에 정점이 있는 원뿔 경계로 정의된 구 혹은 공의 일부입니다.

https://en.wikipedia.org/wiki/Spherical_sector

Albedo

- 정의
 - 물체가 빛을 받았을 때 반사하는 정도를 나타내는 단위임.
- 의미
 - 물체의 재질(texture)과 빛이 반사하는 정도는 관련이 있으며 물체의 표면 재질이 거칠다면, 입사된 빛은 여러 방향으로 불규칙하게 반사될 것임.

빛의 종류

- 빛은 크기 3가지로 추상화 할 수 있음.

- Ambient Light(주변 광, 간접 광)

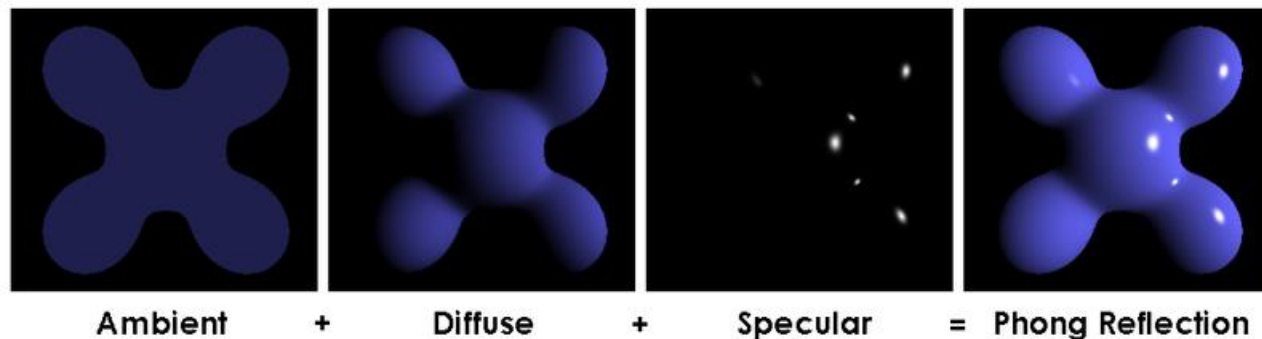
- 수 많은 반사(albedo 와 관련)를 거쳐서 광원이 불분명한 빛.
- Light direction 에 의존하기 않음.
- 물체를 덮고 있는 빛이며, 일정한 밝기와 색으로 표현됨.

- Diffuse Light(분산 광)

- 물체의 표면(normal vector 와 관련)에서 분산되어 눈으로 바로 들어오는 빛.
- 각도에 따라 밝기가 다름.

- Specular Light(반사 광)

- 분산광과 달리 한 방향으로 완전히 반사되는
- 반사되는 부분은 흰색의 광으로 보임.



Ambient + Diffuse + Specular = Phong Reflection

Ambient Light

- 각도 혹은 세기에 상관없이 물체, 배경에 스며들어 일정함.
- 물체가 발하는 평균 색상, 혹은 고정 색상으로 물체가 비치는 특정 빛에 의하여 영향을 거의 받지 않고 vertex 에 대해서 동일한 강도를 가짐. 즉, 물체의 모든 면에 닿는 강도가 같고 방향이 없음.
- $I = k_a I_a$
 - 여기서, k_a 는 표면으로 부터 반사되는 빛의 비율(ambient 반사 계수, 물체의 색으로 주로 고정), I_a 는 반사되는 빛의 밝기(강함 정도, 주변광의 세기) 임.

물체가 빨간 색이고 RGB (1,0,0) 주변
광의 세기가 0.1일 때, 최종 색

`final color = material color * ambient light color`
(반사계수, 즉 물체 색) (주변광의 세기)

`final color = {1, 0, 0} * {0.1, 0.1, 0.1} = {0.1, 0.0, 0.0}`

Diffuse Light – Lambert Cosine Law

- 하나의 **방향성이 있는 빛에 의한 물체 표면의 색깔임.**
- 이는 **빛의 방향 L** (normal vector)과 물체의 Normal vector N 와의 관계($\cos\theta = N \cdot L$)에 따라 색상의 강도가 달라짐. => Lambert factor (범위 $[0,1]$)
 - 표면(normal vector)의 방향과 빛의 방향이 같은 방향을 향할 때, 가장 밝고 정 반대일때 가장 어두움.

```
light vector = light position - object position  
cosine = dot product(object normal, normalize(light vector))  
lambert factor = max(cosine, 0)
```

- 여기에, 계수(빛이 얼마나 반사하는지에 대한 정도) k_d 와 조명의 밝기 I_l 를 곱함.

- $I = k_d I_l \cos\theta = k_d I_l (N \cdot L)$

한 점에서 빛은 멀어질수록 같은 면적에 따른 빛의 세기가 약해진다. (역제곱 법칙)

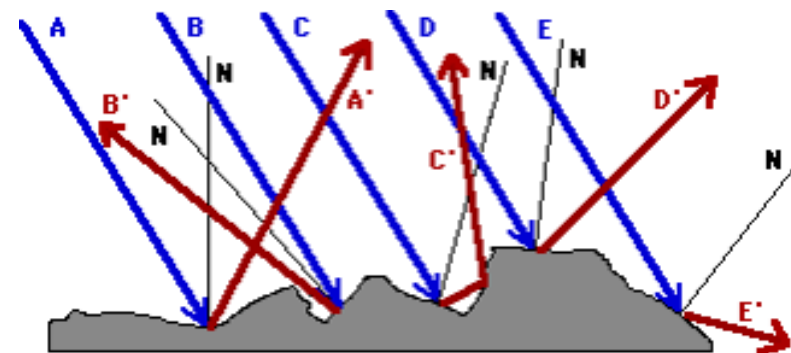
$\text{luminosity} = 1 / (\text{distance} * \text{distance})$

거리가 1보다 작을때를 위해 식을 수정하면

$\text{luminosity} = 1 / (1 + (\text{distance} * \text{distance}))$

최종적으로 분산광에 의한 물체 색이다.

$\text{final color} = \text{material color} * (\text{light color} * \text{lambert factor} * \text{luminosity})$



Lambert Light Model

- $I = k_a I_a + k_d I_l (N \cdot L)$
- 이 Diffuse Color와 Ambient Color가 합쳐지면 우리가 평상 시에 보는 3D 공간에서의 물체의 색깔과 거의 흡사하게 됨.
- 실제로, 저자들은 다음과 같이 diffusion-term 에 최대·최소의 범위를 정함.

$$J_{uv} = (k_s + k_d \max\{0, \langle l, n_{uv} \rangle\}) \cdot a_{uv}$$

↑
Ambient-term

↑
Diffusion-term

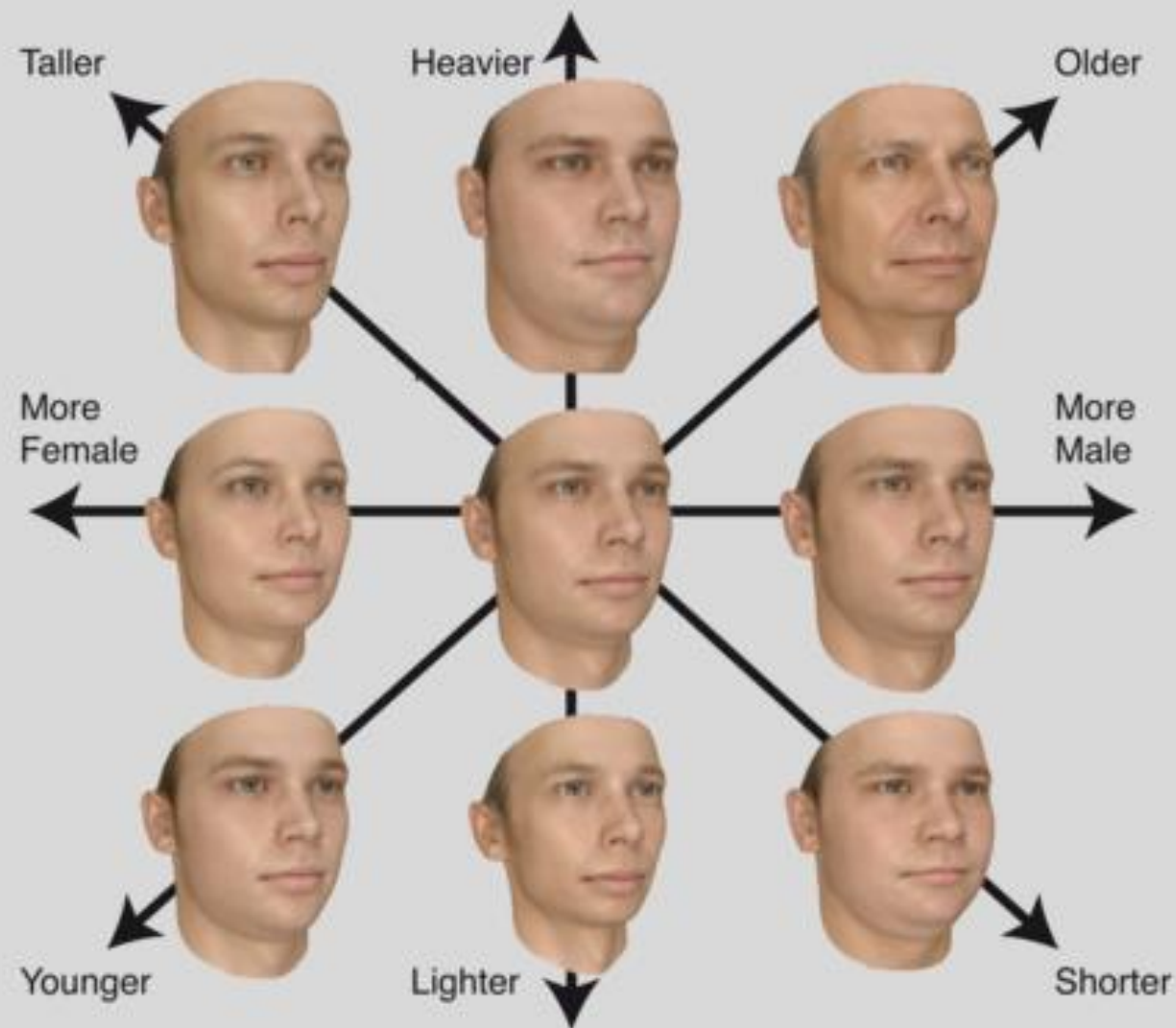
Method – Image formulation model

Perceptual loss

- L1 loss 함수는 작은 기하학적인 불일치(geometric imperfections) 에도 민감하게 반응하여, 복원된 결과 영상을 blurry 하게 만든다는 단점 이 존재함.
- 이를 극복하기 위하여, perceptual loss 를 추가함.
 - $\mathcal{L}_p^{(k)}(\hat{I}, I, \sigma) = -\frac{1}{|\Omega|} \log \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\sqrt{2}(e^{(k)}(\hat{I}) - e^{(k)}(I))^2}{2\sigma^2}}$
: gaussian distribution(L_2).
 - 여기서, $e^{(k)}(I)$ 는 입력영상, I 에 대한 pre-trained encoder 의 feature(representation) 임.
 - 이때, 사용된 features 는 relu3_3 layer 의 출력 feature 임.
- perceptual loss 를 고려한 모델의 최종 손실 함수는 다음과 같음.
 - $\mathcal{L} + \lambda_p \mathcal{L}_p$
 - 여기서, $\lambda_p = 1$ 로 설정.

Attributes

The training data was labelled with gender, height, weight, and age. By varying face coefficients along the directions of maximal variance for an attribute as observed in the training data, it is possible to systematically manipulate these attributes. We provide the directions of maximal variance in the file 04_attributes.mat



논문 작성시, 참고할 만한 문장

