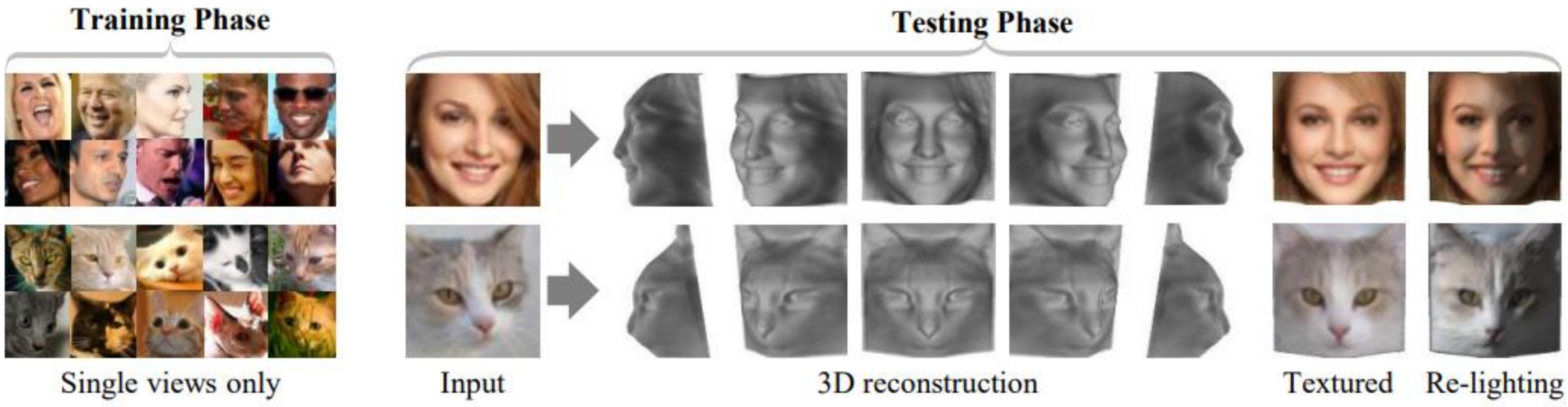
Unsupervised Learning of Probably Symmetric Deformable 3D Objects from Images in the Wild

CVPR2020 Best paper

20.08.12 Leeminsoo

DAVIAN
Data and Visual Analytics Lab

Introduction



- This paper proposes a method to learn 3D deformable object categories from raw single-view images, without external supervision.
- The method is based on an autoencoder that factors each input image into depth, albedo, viewpoint and light direction.
- In order to disentangle these components without supervision, it uses the fact that many object categories have a symmetric structure.

Demo

https://elliottwu.com/projects/unsup3d/

Background Knowledge

Depth Map

Normal Map

Shading

Albedo

Canonical View

Orginal Img

Photo Geometric Autoencoding

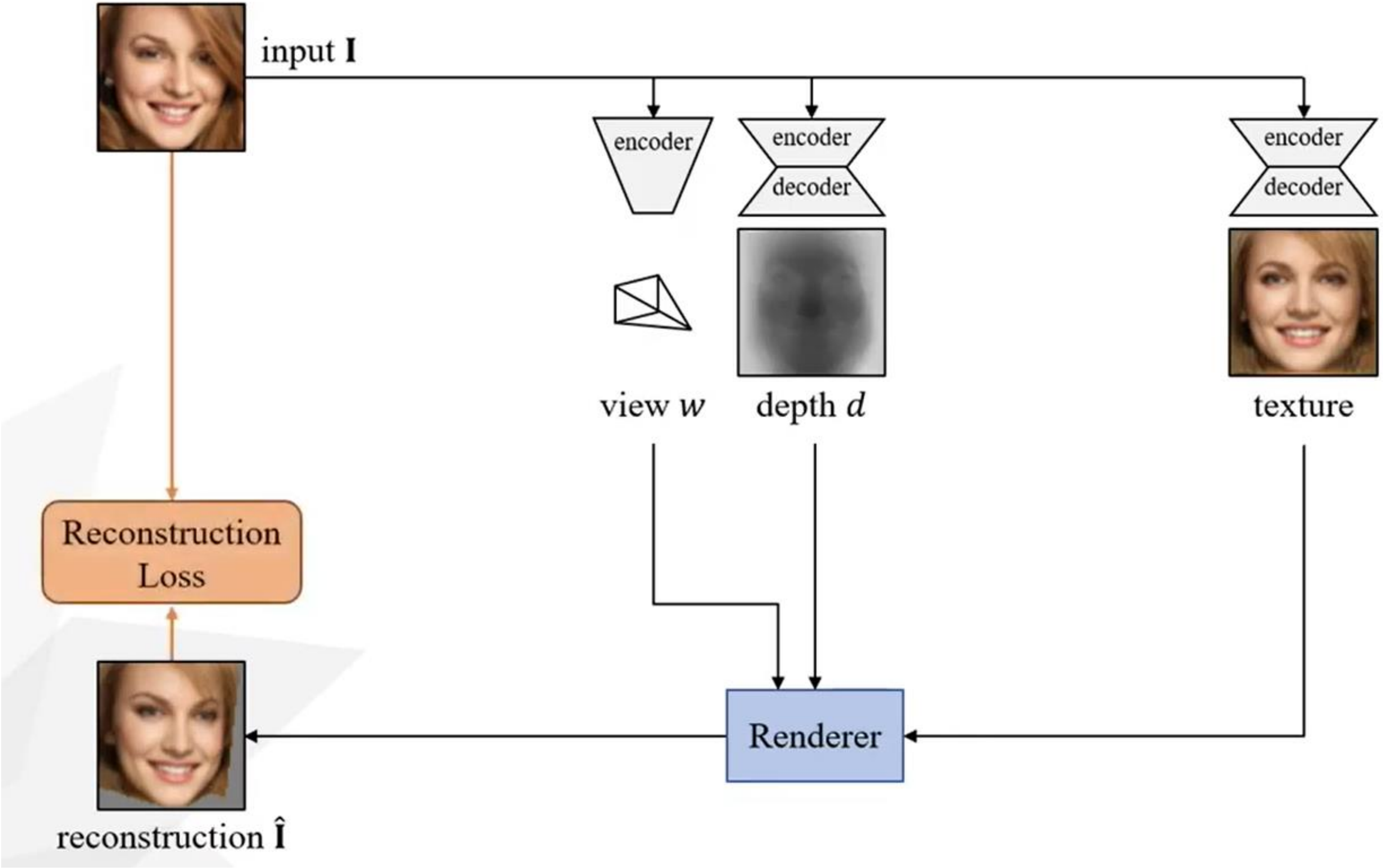


Photo Geometric Autoencoding

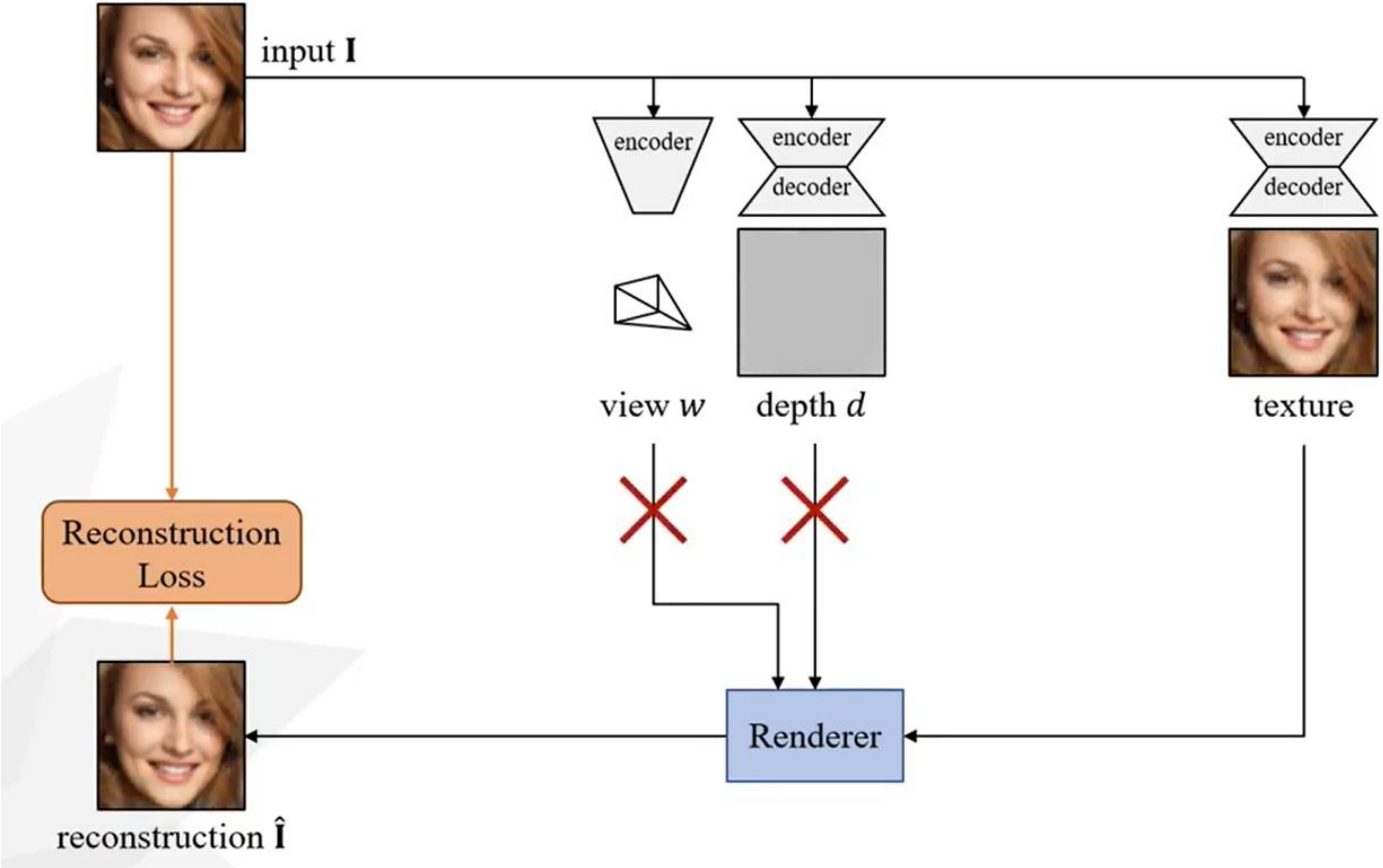


Photo Geometric Autoencoding

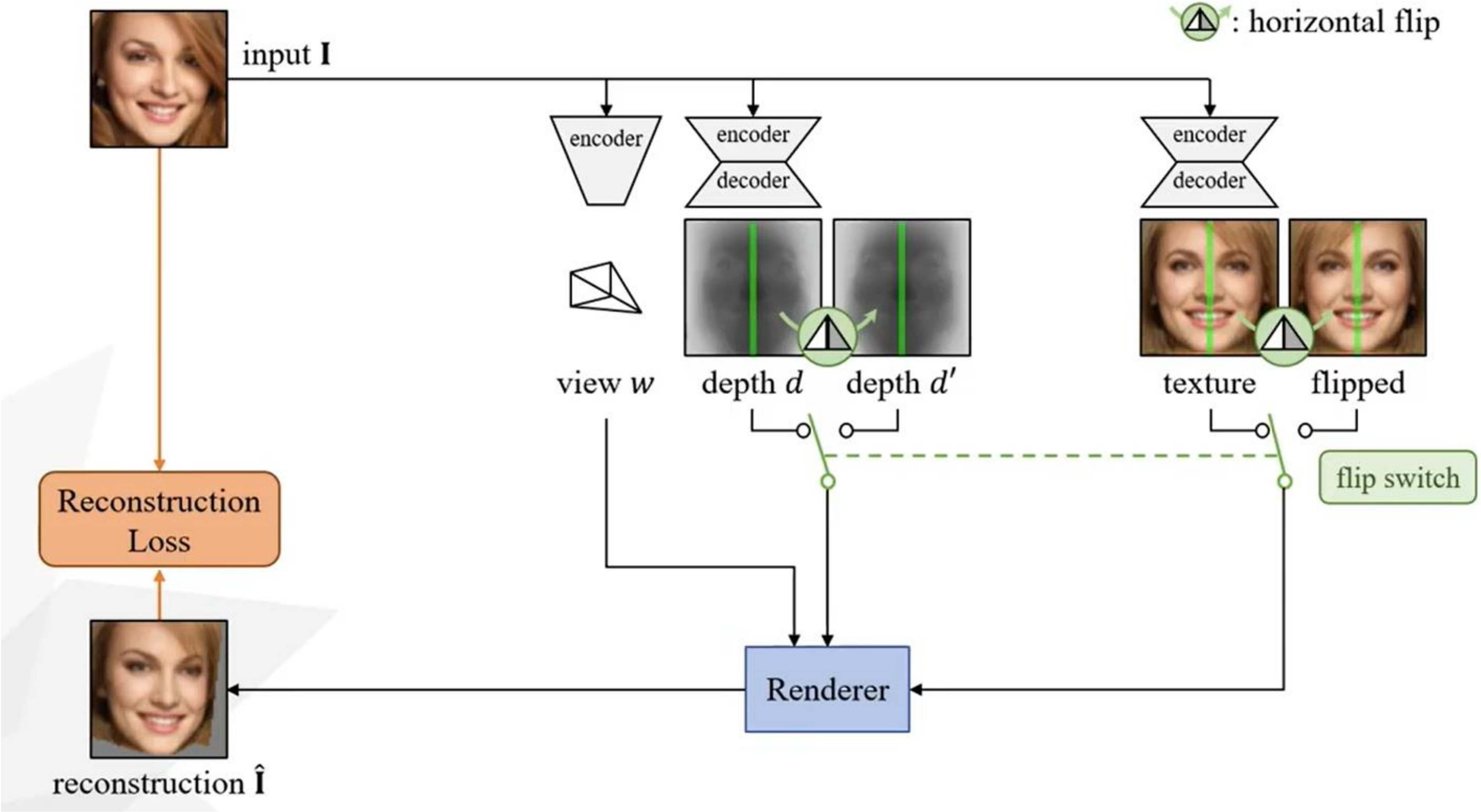


Photo Geometric Autoencoding

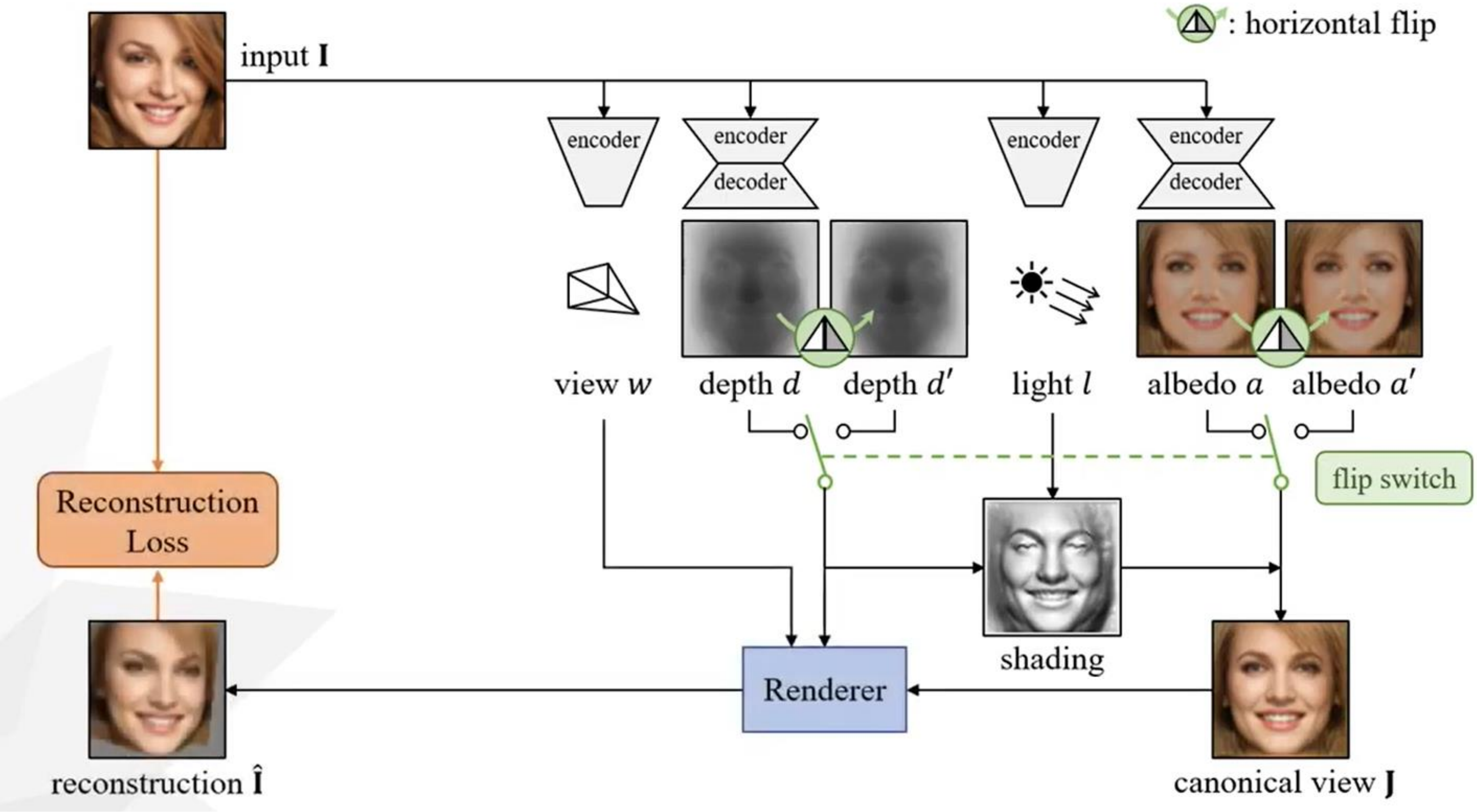
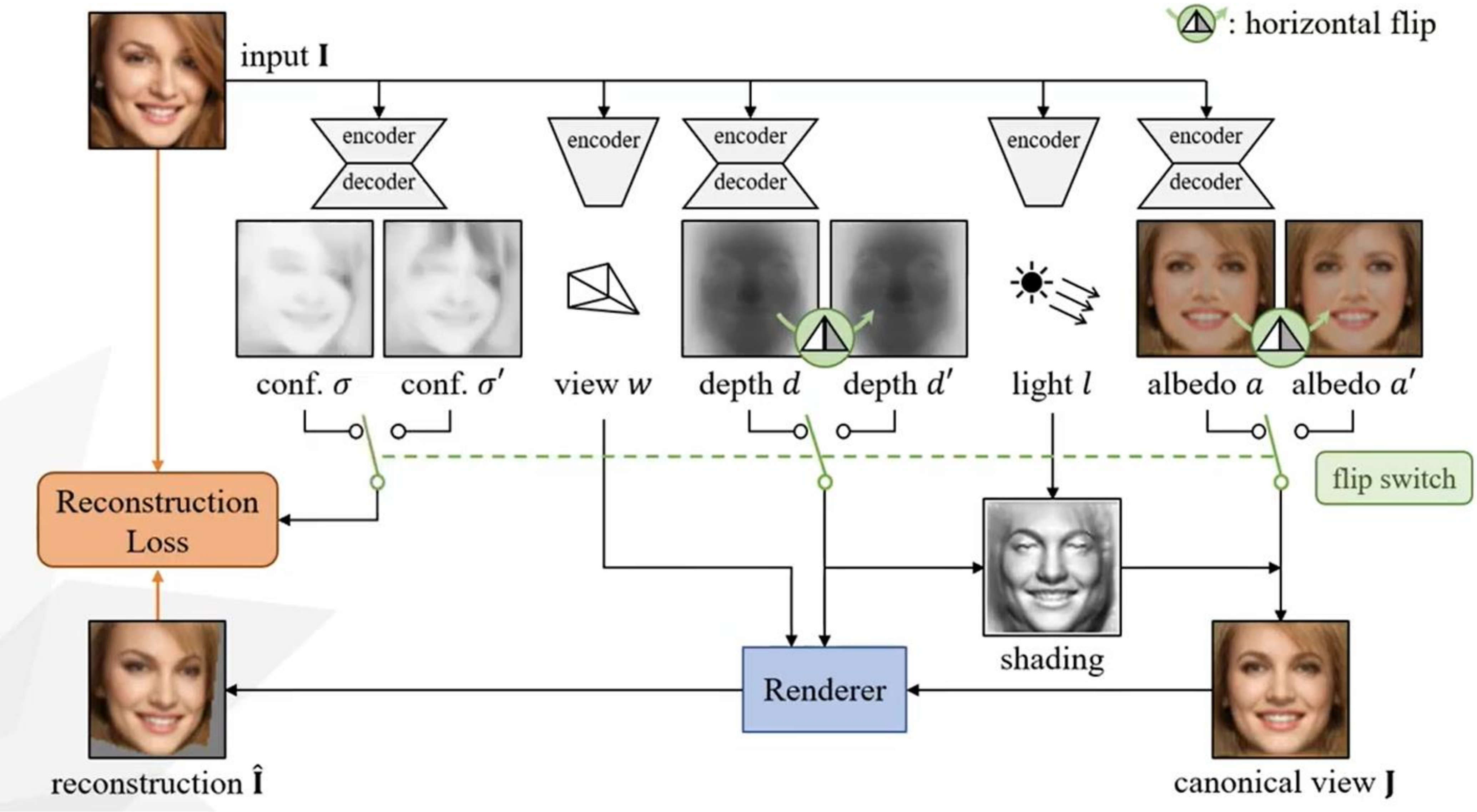
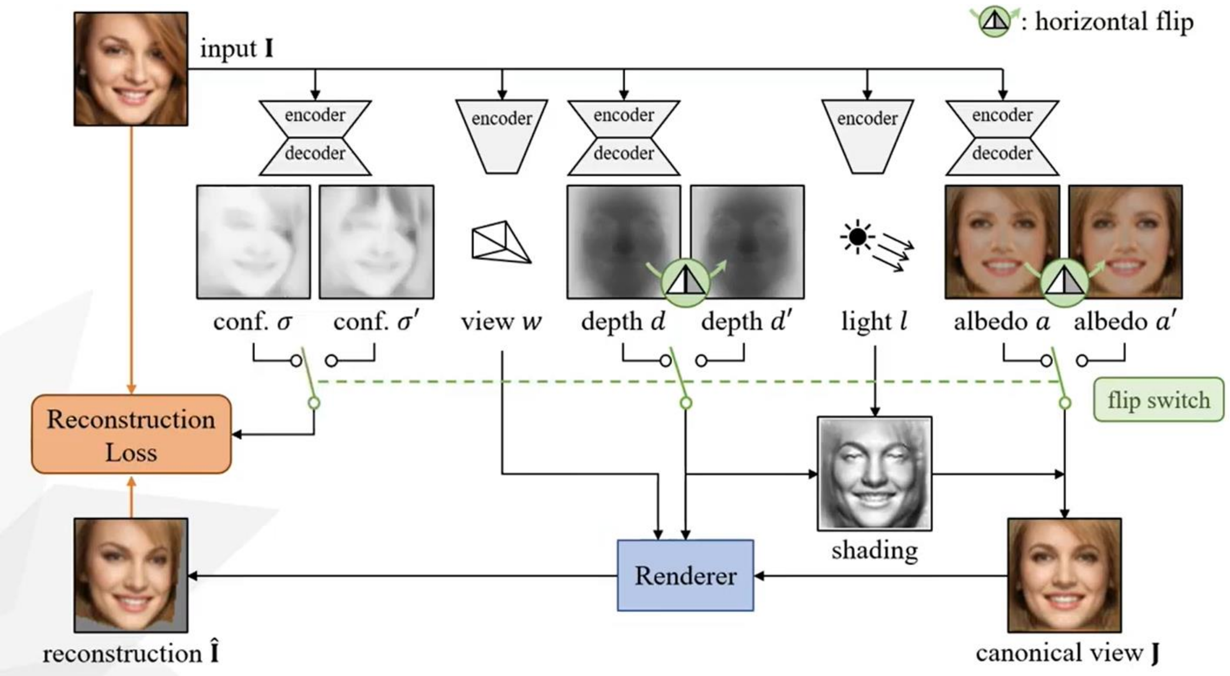


Photo Geometric Autoencoding



Detail – Confidence Map and Loss Function



$$\Phi(\mathbf{I}) = (d, a, w, l, \sigma, \sigma')$$

$$d : \Omega \rightarrow \mathbb{R}_+ \quad a : \Omega \rightarrow \mathbb{R}^3$$

$$\textit{light } l \in \mathbb{R}^4 \quad \textit{viewpoint } w \in \mathbb{R}^6$$

$$\hat{\mathbf{I}} = \Pi\left(\Lambda(a, d, l), d, w\right)$$

$$\hat{\mathbf{I}}' = \Pi\left(\Lambda(a', d', l), d', w\right), \quad a' = \operatorname{flip} a, \quad d' = \operatorname{flip} d.$$

$$\mathcal{L}(\hat{\mathbf{I}}, \mathbf{I}, \sigma) = -\frac{1}{|\Omega|} \sum_{uv \in \Omega} \ln \frac{1}{\sqrt{2}\sigma_{uv}} \exp -\frac{\sqrt{2}\ell_{1,uv}}{\sigma_{uv}},$$

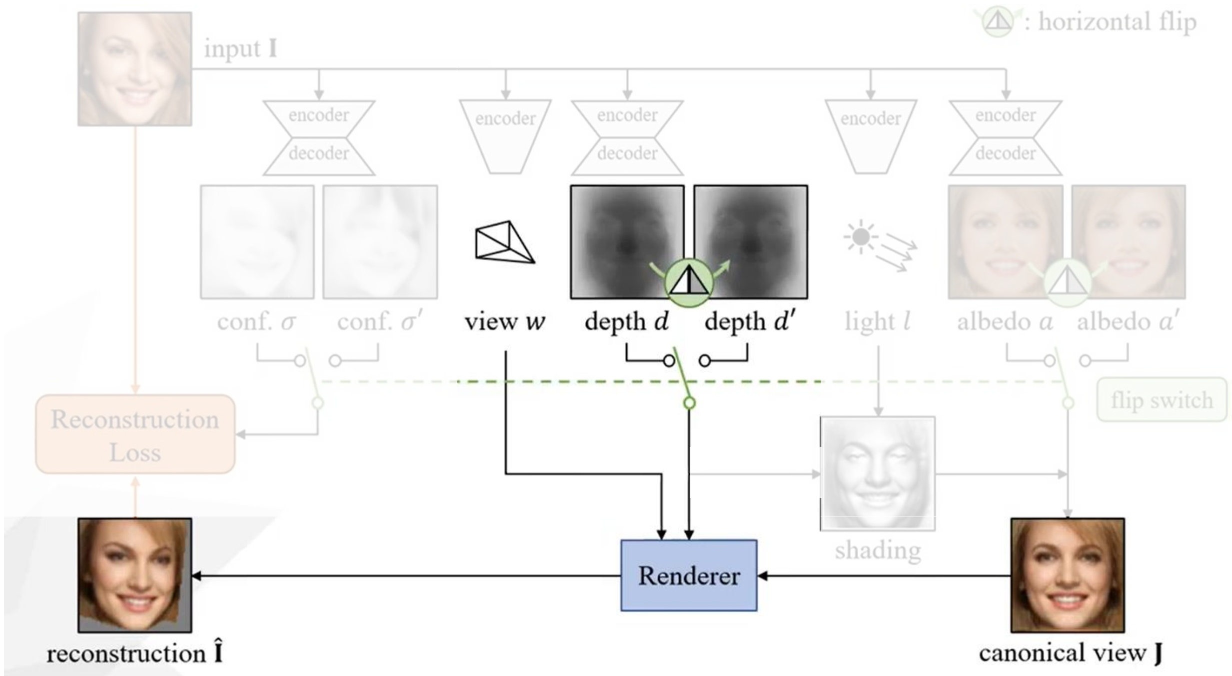
$$\mathcal{E}(\Phi; \mathbf{I}) = \mathcal{L}(\hat{\mathbf{I}}, \mathbf{I}, \sigma) + \lambda_{\mathrm{f}} \mathcal{L}(\hat{\mathbf{I}}', \mathbf{I}, \sigma'),$$

- Assuming that depth and albedo, which are reconstructed in a canonical frame, are symmetric, the model can discover a canonical view for the object.

- By obtaining a second reconstruction $\hat{I}'$ from the flipped depth and albedo, the model effectively incorporate the symmetry constraint in the depth and albedo.

- σ' can assign a higher reconstruction uncertainty where the symmetry assumption is not satisfied

Detail – Image formation model : Transformation



$P = (P_x, P_y, P_z) \in \mathbb{R}^3 \qquad p = (u, v, 1)$

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

$$P = d_{uv} \cdot K^{-1} p.$$

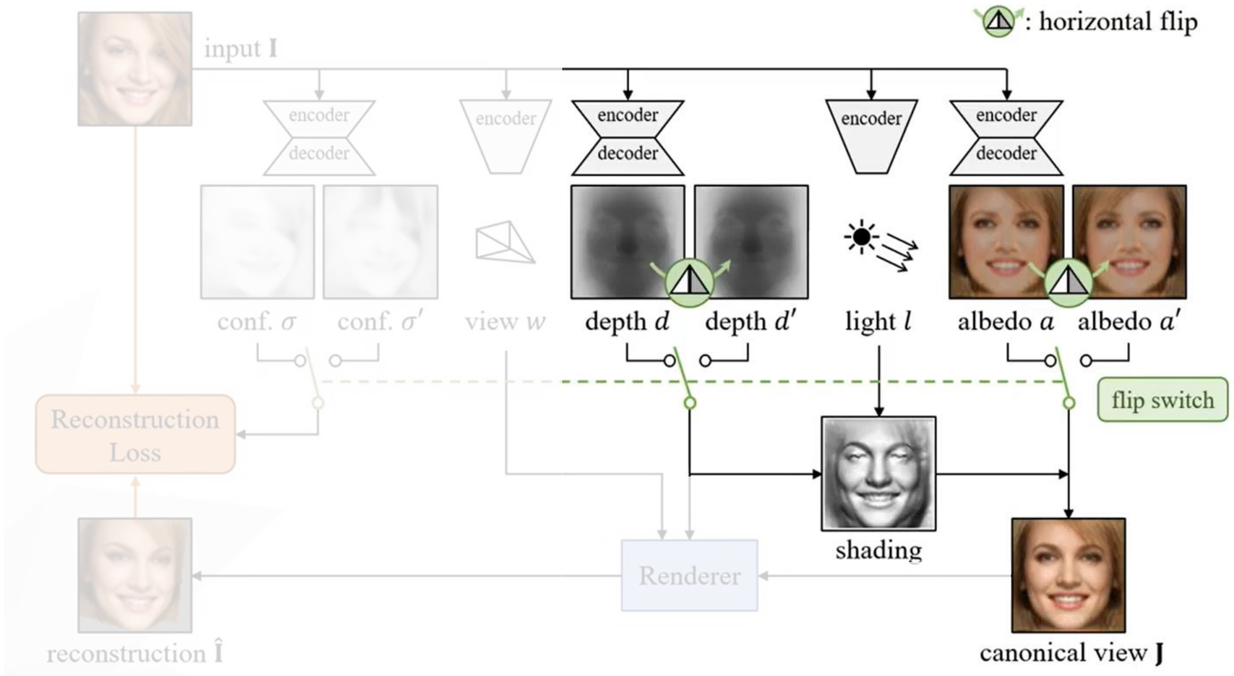
warping function $\eta_{d,w} : (u, v) \mapsto (u', v')$ given by:

$$p' \propto K(d_{uv} \cdot RK^{-1}p + T),$$

where $p' = (u', v', 1)$.

- The (R,T) transforms 3D points from the canonical view to the actual view, thus a pixel (u, v) in canonical view is mapped to the pixel (u', v') in the actual view.
- The reprojection function Π takes as input the depth d and the viewpoint w and applies the resulting warp to the canonical image J to obtain the reconstruction image $\hat{I} = \Pi(J, d, w)$.

Detail – Image formation model : Canonical view



$$t^u_{uv} = d_{u+1,v} \cdot K^{-1}(p+e_x) - d_{u-1,v} \cdot K^{-1}(p-e_x)$$

where p is defined above and $e_x = (1,0,0)$.

$$n_{uv} \propto t^u_{uv} \times t^v_{uv}.$$

$$l = (l_x, l_y, 1)^T/(l_x^2 + l_y^2 + 1)^{0.5}$$

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

- Given depth map d, we drive the normal map n by associating to each pixel (u, v) a vector normal to the underlying 3D surface.
- The normal n_{uv} is multiplied by the light direction l to obtain a value for the directional illumination and the latter is added to the ambient light.
- The result is multiplied by the albedo to obtain the illuminated texture.

Experiment – Ablation study

No	Method	SIDE ($\times 10^{-2}$) ↓	MAD (deg.) ↓
(1)	Ours full	0.793 $\pm$0.140	16.51 $\pm$1.56
(2)	w/o albedo flip	2.916 $\pm$0.300	39.04 $\pm$1.80
(3)	w/o depth flip	1.139 $\pm$0.244	27.06 $\pm$2.33
(4)	w/o light	2.406 $\pm$0.676	41.64 $\pm$8.48
(5)	w/o perc. loss	0.931 $\pm$0.269	17.90 $\pm$2.31
(6)	w/ self-sup. perc. loss	0.815 $\pm$0.145	15.88 $\pm$1.57
(7)	w/o confidence	0.829 $\pm$0.213	16.39 $\pm$2.12

Table 3: **Ablation study.**

$$E_{\mathrm{SIDE}}(\bar{d}, d^*) = \left(\frac{1}{WH}\sum_{uv}\Delta_{uv}^2 - \left(\frac{1}{WH}\sum_{uv}\Delta_{uv}\right)^2\right)^{\frac{1}{2}}$$

- In (2), the albedo is not encouraged to be symmetric in the canonical space, which fails to canonicalize the viewpoint of the object and to use cues from symmetry to recover shape.

- In (4), the model predicts a shading map instead of computing it from depth and light direction. This also harms performance significantly.

- In (7), The accuracy does not drop significantly, but it's because faces in BFM are highly symmetric(do not have hair).

Experiment – Ablation study

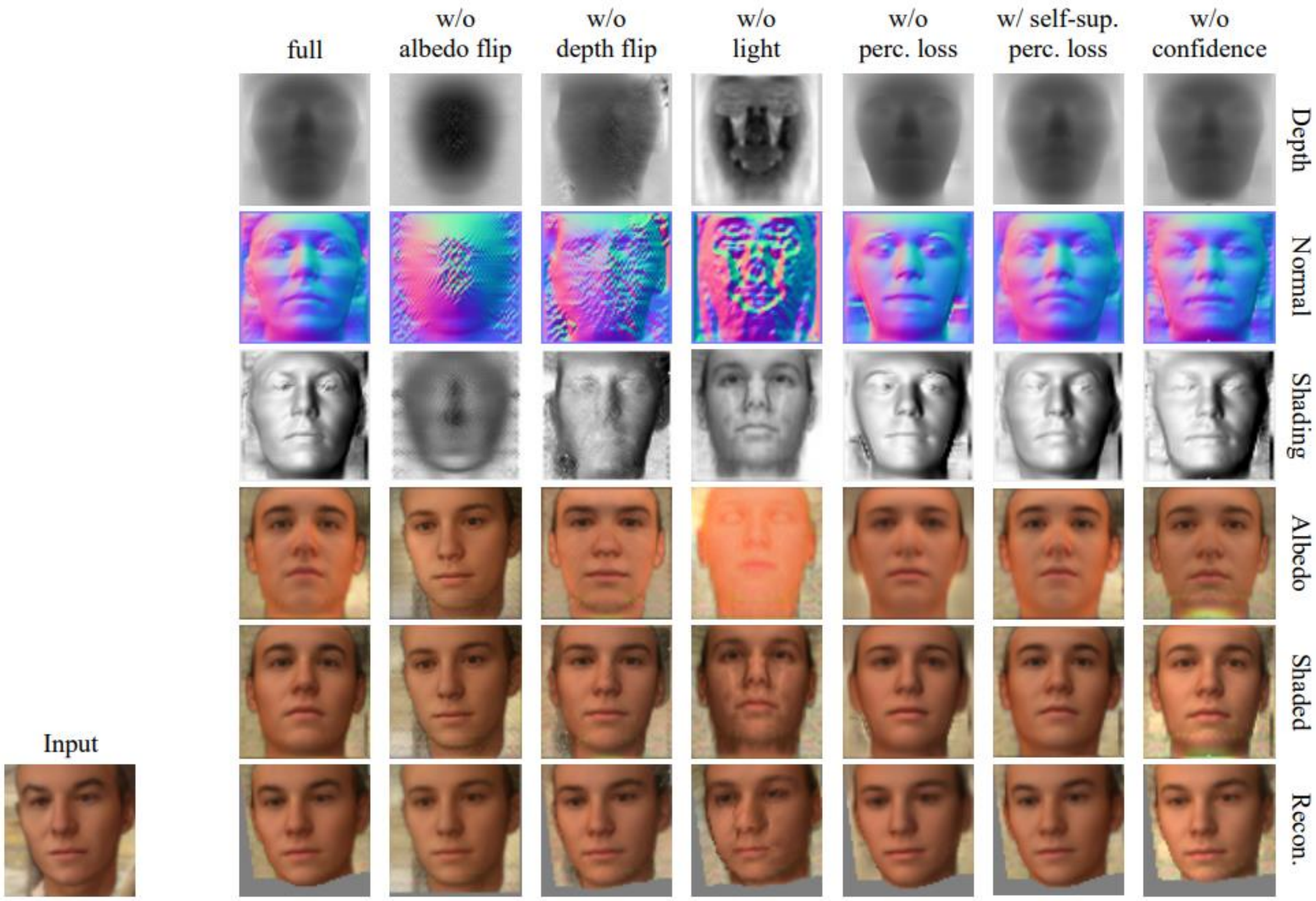
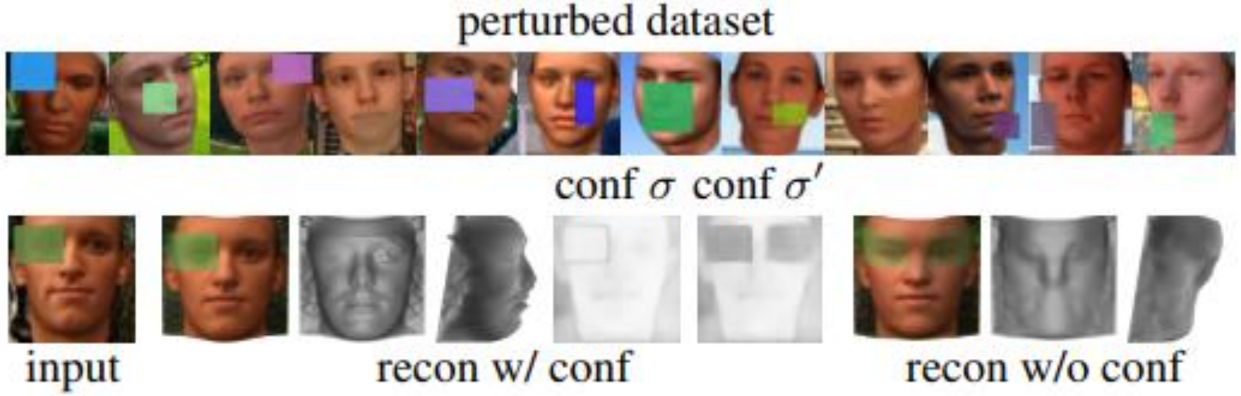


Figure 9: **Qualitative results of the ablated models.**

Experiment – Asymmetric perturbation

	SIDE ($\times 10^{-2}$) $\downarrow$	MAD (deg.) $\downarrow$
No perturb, no conf.	0.829 $\pm$0.213	16.39 $\pm$2.12
No perturb, conf.	0.793 $\pm$0.140	16.51 $\pm$1.56
Perturb, no conf.	2.141 $\pm$0.842	26.61 $\pm$5.39
Perturb, conf.	0.878 $\pm$0.169	17.14 $\pm$1.90

Table 4: **Asymmetric perturbation.**



- In order to demonstrate the uncertainty modelling allows the model to handle asymmetry, the authors add asymmetric perturbations to BFM.
- Without the confidence maps, the model always predicts a symmetric albedo and geometry reconstruction often fails.
- With the confidence estimates, the model is able to reconstruct the asymmetric faces correctly.

Experiment – Qualitative results

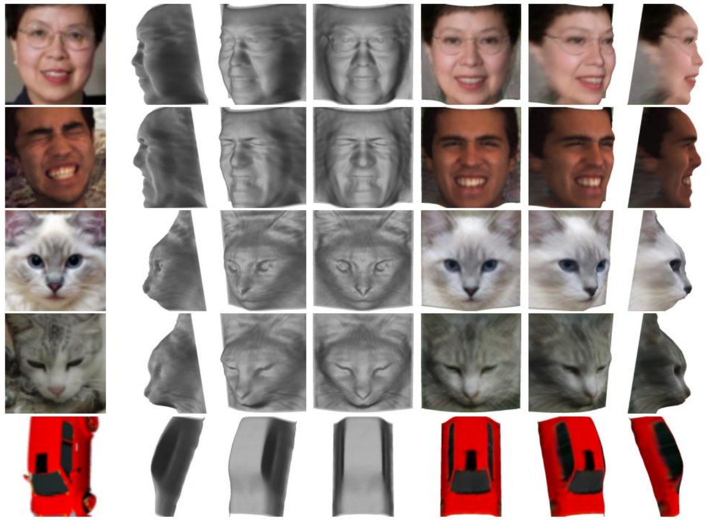


Figure 4: **Reconstruction of faces, cats and cars.**

Figure 5: **Reconstruction of faces in paintings.**

Figure 6: **Symmetry plane and asymmetry detection.** (a): our model can reconstruct the "intrinsic" symmetry plane of an in-the-wild object even though the appearance is highly asymmetric. (b): asymmetries (highlighted in red) are detected and visualized using confidence map σ'.

- The reconstructed 3D face contain fine details of the nose, eyes and mouth even in the presence of extreme facial expression.
- Since the model predicts a canonical view of the objects that is symmetric, we can easily find the symmetry plane, which is otherwise non-trivial to detect from in-the-wild images.
- Overlaying the predicted σ' onto the image, the model can detect asymmetric regions.

Experiment – Comparison with other models

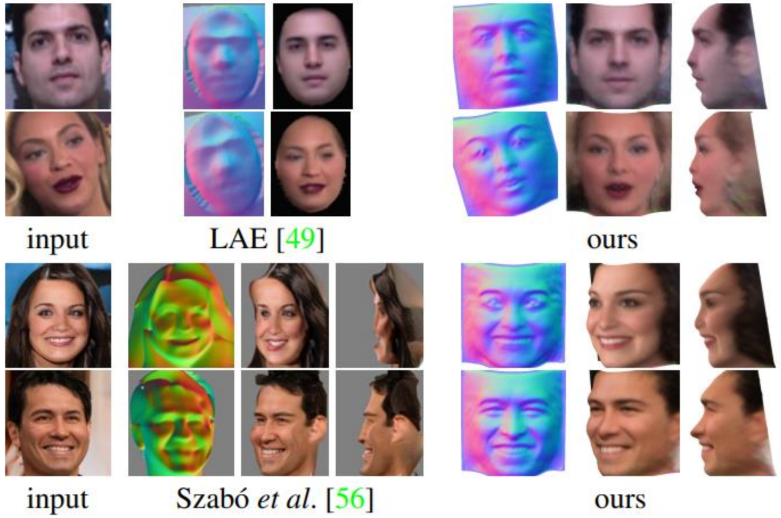


Figure 7: **Qualitative comparison to SOTA.** Our method recovers much higher quality shapes compared to [49, 56].

[49] Unsupervised learning of a fully-disentangled 3d morphable model using deep non-figid structure from motion. ICCV Workshop 2019

[56] Unsupervised generative 3d shape learning from natural images. arXiv 2019

	Depth Corr. ↑
Ground truth	66
AIGN [61] (**supervised**, from [40])	50.81
DepthNetGAN [40] (**supervised**, from [40])	58.68
MOFA [57] (**model-based**, from [40])	15.97
DepthNet [40] (from [40])	26.32
DepthNet [40] (from GitHub)	35.77
Ours	48.98
Ours (w/ CelebA pre-training)	54.65

Table 5: **3DFAW keypoint depth evaluation.**

Appendix

Encoder	Output size
Conv(3, 32, 4, 2, 1) + ReLU	32
Conv(32, 64, 4, 2, 1) + ReLU	16
Conv(64, 128, 4, 2, 1) + ReLU	8
Conv(128, 256, 4, 2, 1) + ReLU	4
Conv(256, 256, 4, 1, 0) + ReLU	1
Conv(256, c_{out}, 1, 1, 0) + Tanh $\rightarrow$ *output*	1

Table 7: Network architecture for viewpoint and lighting. The output channel size c_{out} is 6 for viewpoint, corresponding to rotation angles $w_{1:3}$ and translations $w_{4:6}$ in x, y and z axes, and 4 for lighting, corresponding to k_s, k_d, l_x and l_y.

Encoder	Output size
Conv(3, 64, 4, 2, 1) + GN(16) + LReLU(0.2)	32
Conv(64, 128, 4, 2, 1) + GN(32) + LReLU(0.2)	16
Conv(128, 256, 4, 2, 1) + GN(64) + LReLU(0.2)	8
Conv(256, 512, 4, 2, 1) + LReLU(0.2)	4
Conv(512, 256, 4, 1, 0) + ReLU	1
Decoder	**Output size**
Deconv(256, 512, 4, 1, 0) + ReLU	4
Conv(512, 512, 3, 1, 1) + ReLU	4
Deconv(512, 256, 4, 2, 1) + GN(64) + ReLU	8
Conv(256, 256, 3, 1, 1) + GN(64) + ReLU	8
Deconv(256, 128, 4, 2, 1) + GN(32) + ReLU	16
Conv(128, 128, 3, 1, 1) + GN(32) + ReLU	16
Deconv(128, 64, 4, 2, 1) + GN(16) + ReLU	32
Conv(64, 64, 3, 1, 1) + GN(16) + ReLU	32
Upsample(2)	64
Conv(64, 64, 3, 1, 1) + GN(16) + ReLU	64
Conv(64, 64, 5, 1, 2) + GN(16) + ReLU	64
Conv(64, c_{out}, 5, 1, 2) + Tanh $\rightarrow$ *output*	64

Table 8: Network architecture for depth and albedo. The output channel size c_{out} is 1 for depth and 3 for albedo.

Encoder	Output size
Conv(3, 64, 4, 2, 1) + GN(16) + LReLU(0.2)	32
Conv(64, 128, 4, 2, 1) + GN(32) + LReLU(0.2)	16
Conv(128, 256, 4, 2, 1) + GN(64) + LReLU(0.2)	8
Conv(256, 512, 4, 2, 1) + LReLU(0.2)	4
Conv(512, 128, 4, 1, 0) + ReLU	1
Decoder	**Output size**
Deconv(128, 512, 4, 1, 0) + ReLU	4
Deconv(512, 256, 4, 2, 1) + GN(64) + ReLU	8
Deconv(256, 128, 4, 2, 1) + GN(32) + ReLU	16
↳ Conv(128, 2, 3, 1, 1) + SoftPlus $\rightarrow$ *output*	16
Deconv(128, 64, 4, 2, 1) + GN(16) + ReLU	32
Deconv(64, 64, 4, 2, 1) + GN(16) + ReLU	64
Conv(64, 2, 5, 1, 2) + SoftPlus $\rightarrow$ *output*	64

Table 9: Network architecture for confidence maps. The network outputs two pairs of confidence maps at different spatial resolutions for photometric and perceptual losses.