Unsupervised Learning of Probably Symmetric Deformable 3D objects

February 17th, 2021

Hyeonsu Do

ELECTRICAL & COMPUTER ENGINEERING
UNIVERSITY OF MICHIGAN

Motivation

- Explain the variability of the natural images
- Improve image understanding in general

Constraints of Input

- No ground truth data
  - Keypoints
  - Segmentation
  - Depth maps
  - Prior knowledge
- Unconstrained collection of single view images
  - Should not require multiple views of same instance

Why not Perfect Symmetry

- Why not Perfect Symmetry
  - Variation of poses
  - Albedo
  - Illumination
- Solutions
  - Explicit modeling of illumination
  - Augmenting the model to reason about the lack of symmetry in input images

Main Method

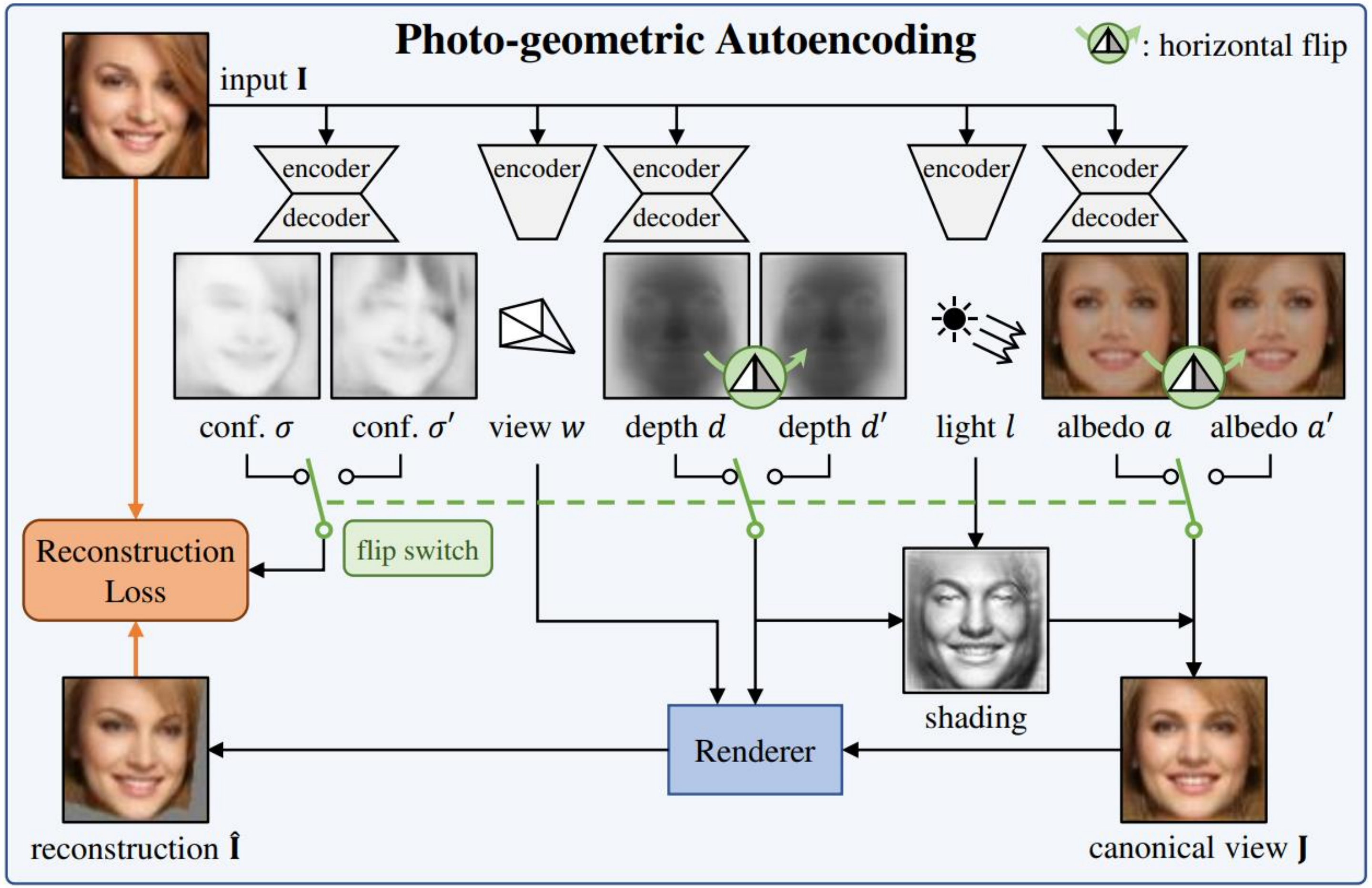


Photo-geometric Autoencoding
: horizontal flip
input I
encoder
decoder
encoder
encoder
decoder
encoder
encoder
decoder
conf. σ
conf. σ′
view w
depth d
depth d′
light l
albedo a
albedo a′
flip switch
Reconstruction Loss
shading
Renderer
reconstruction Î
canonical view J

Network for Viewpoint and Lighting

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

- Output
  - Viewpoint : 6 Channel
  - Lighting : 4 Channel

Network for Depth and 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, 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

- Output
  - Depth : 1 Channel
  - Albedo : 3 Channel

Network for Confidence Map

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

- Output
  - Two pairs of confidence maps
  - Different spatial resolution
  - Photometric and perceptual loss

Photo-geometric autoencoding

- Asymmetric illumination
- Separation of Albedo and Lighting

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

Probably symmetric Objects

- Symmetry implied by the albedo and depth
- Learning objective by combination of the two reconstruction errors, with weighing factor λ

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

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

Image Formation Model

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

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

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

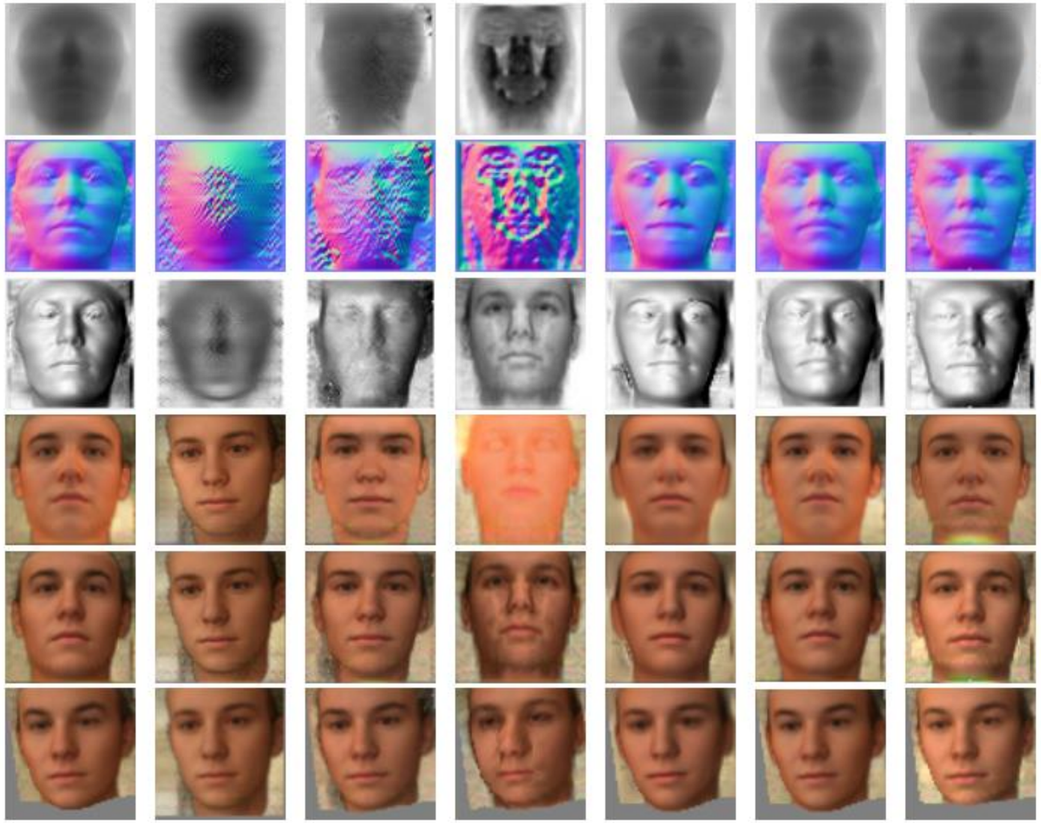
$$\mathbf{J} = \Lambda(a, d, l)$$

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

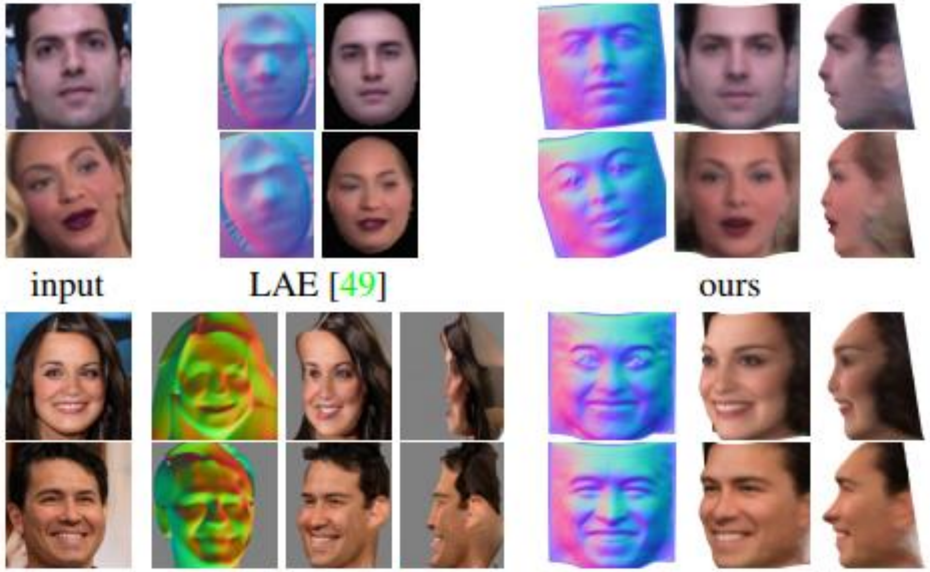
$$l = (l_x, l_y, 1)^T / (l_x^2 + l_y^2 + 1)^{0.5}$$

Results – Ablated Models

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



Results – Compared with SOTA

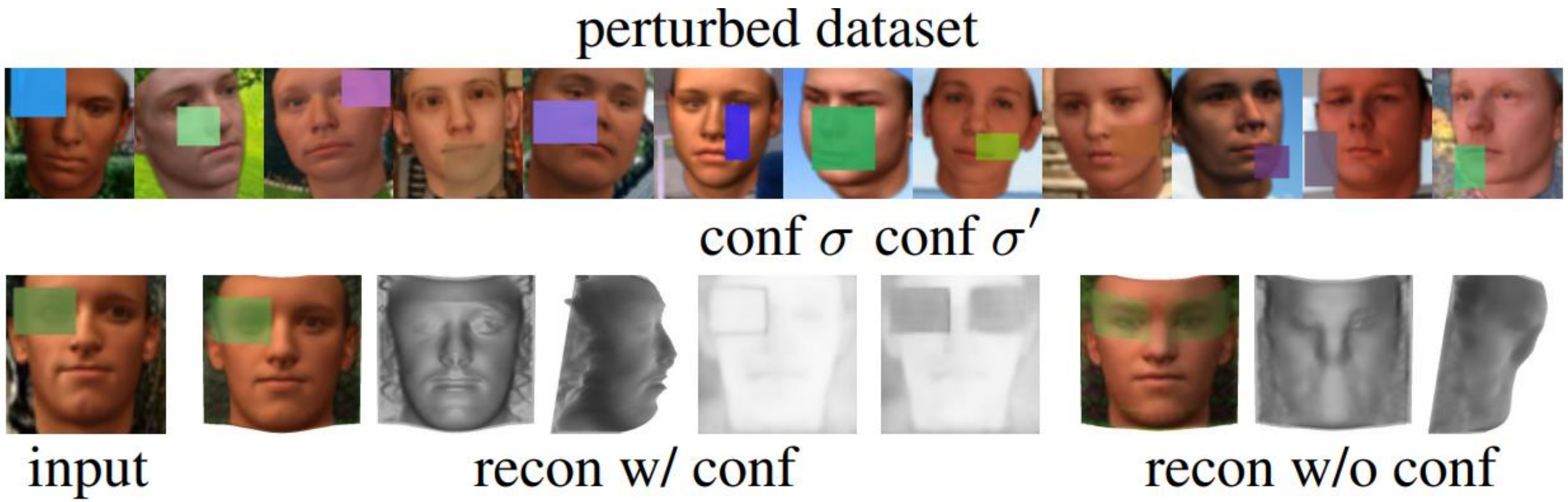


	Depth Corr. ↑
Ground truth	66
AIGN [61] (**supervised**, from [40])	50.81
DepthNetGAN [40] (**supervised**, from [40])	58.68
MOFA [57] (**model-based**, from [40])	15.97
DepthNet [40] (from [40])	26.32
DepthNet [40] (from GitHub)	35.77
Ours	48.98
Ours (w/ CelebA pre-training)	54.65

No	Baseline	SIDE ($\times 10^{-2}$) ↓	MAD (deg.) ↓
(1)	Supervised	0.410 ± 0.103	10.78 ± 1.01
(2)	Const. null depth	2.723 ± 0.371	43.34 ± 2.25
(3)	Average g.t. depth	1.990 ± 0.556	23.26 ± 2.85
(4)	Ours (unsupervised)	0.793 ± 0.140	16.51 ± 1.56

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

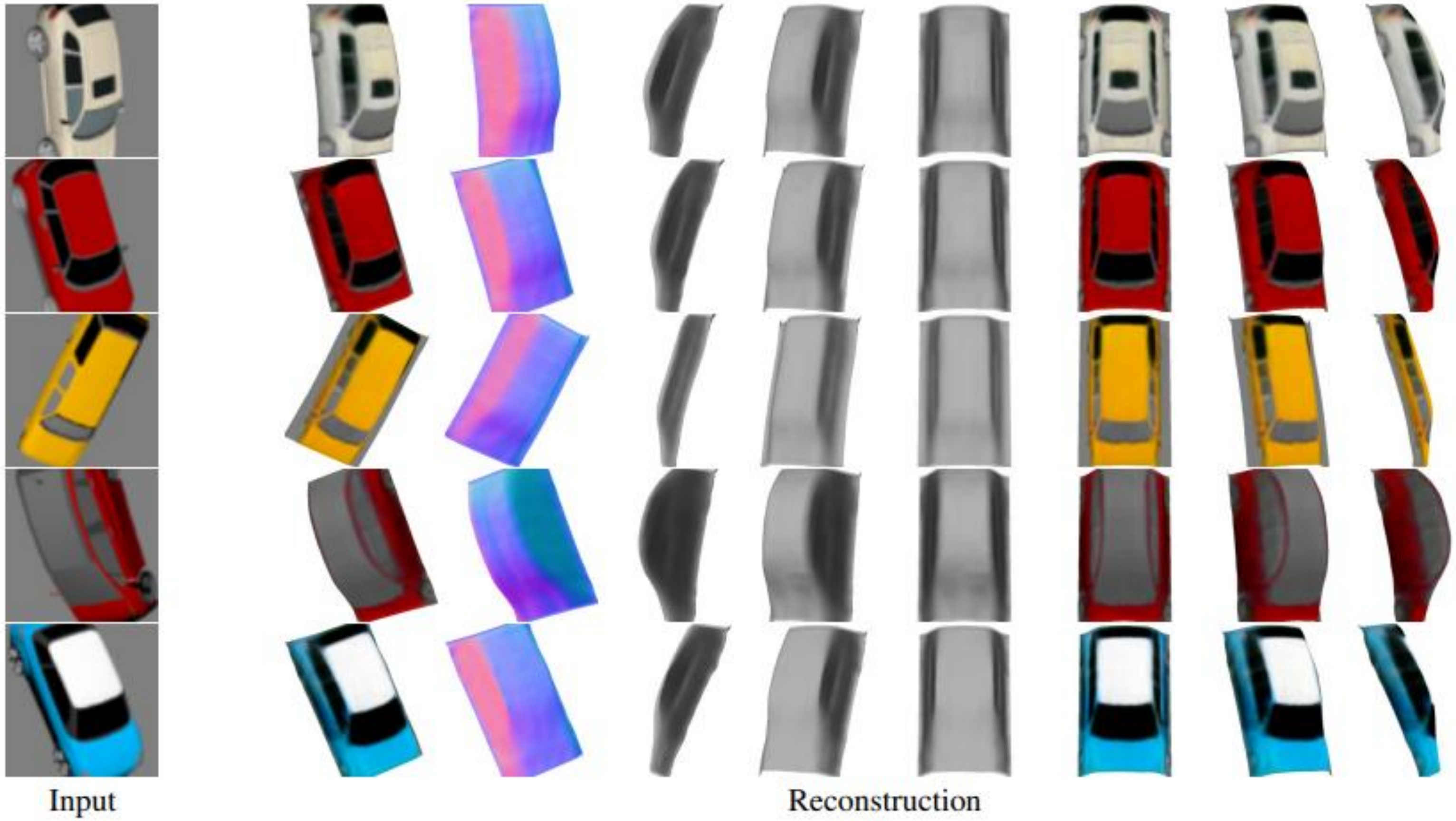


Results – Fail Cases

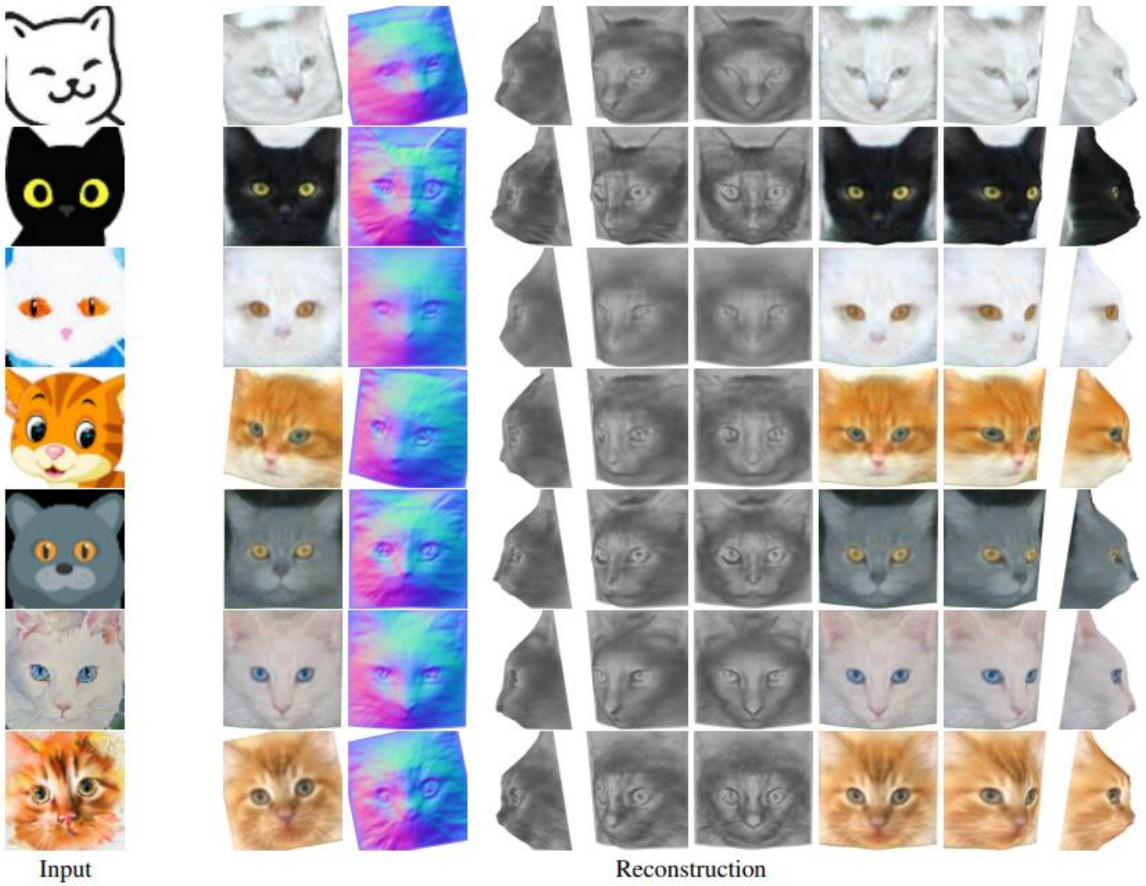
- Extreme Lighting
  - Simple Lambertian Shading Model
- Noisy Texture
  - Dark, noisy textures
- Extreme Pose
  - Poor supervisory signal from reconstruction loss of side images

Results – Re-lighting Effects

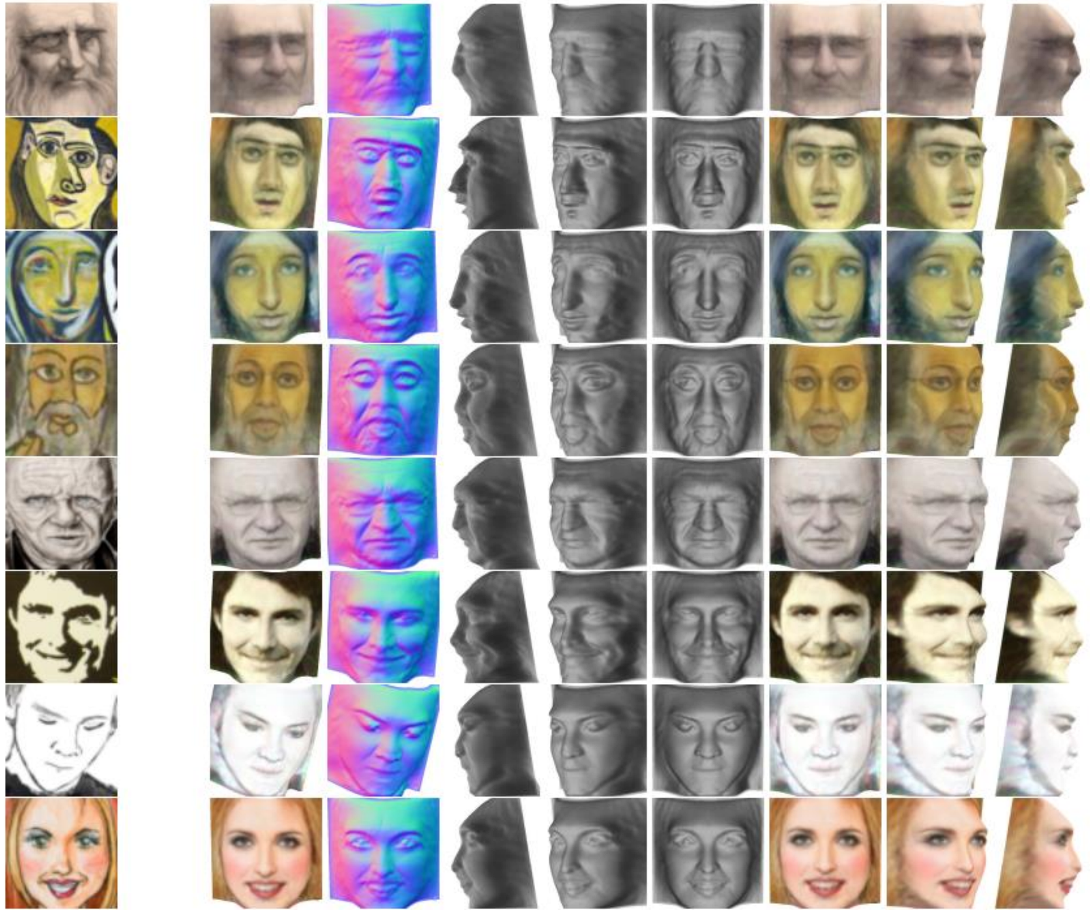
Results – Model vehicles

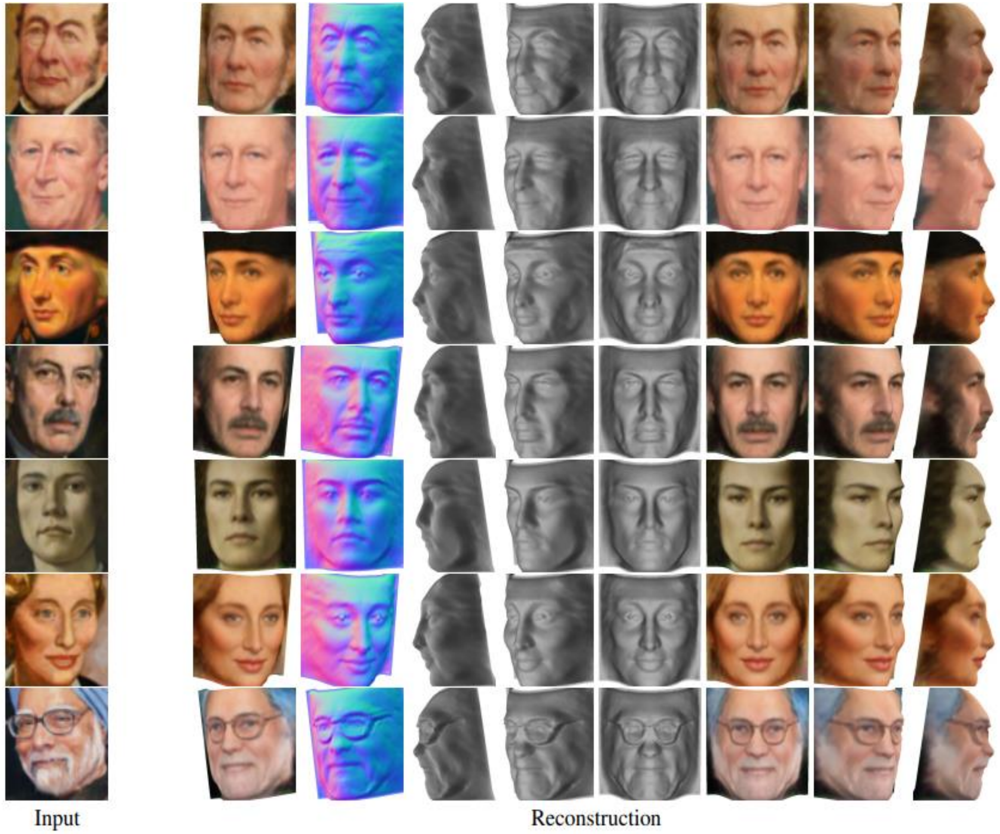


Results – Abstract Drawing(Cats)



Results – Abstract Drawing(human)





Discussion

- This work is very limited by its architecture, requiring symmetry. What ways could the work be modified to be expanded to non-symmetric objects?
- What are some of the applications of this method?
- Is it a large advantage to have non-labeled data to train over supervised training methods?
- Extending to multiple canonical views using a single set of image for complex objects?