Single-Shot Analysis of Refractive Shape Using Convolutional Neural Networks

Jonathan Dyssel Stets,[1] Zhengqin Li,[2] Jeppe Revall Frisvad,[1] Manmohan Chandraker[2]

[1]Technical University of Denmark, [2]University of California, San Diego

IEEE 2019 Winter Conference on Applications of Computer Vision

Rendering a Dataset with Transparent Objects

- A transparent object takes its appearance not only from its backdrop, but from all its surroundings.

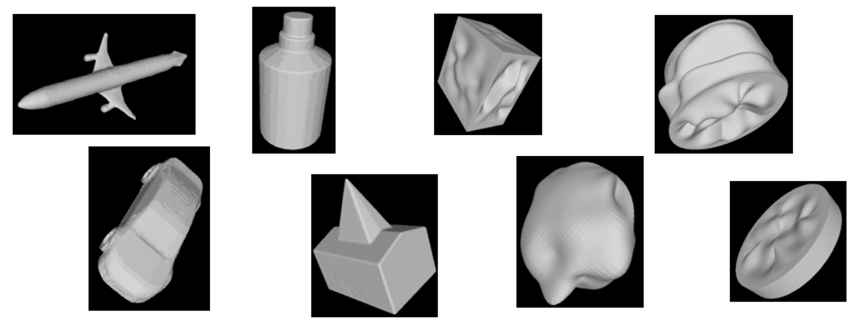
Note the reflected and refracted lamps not seen in the backdrop.

- A validated technique for photorealistic rendering of transparent objects is available.

[Stets, Dal Corso et al. Scene reassembly after multimodal digitization and pipeline evaluation using photorealistic rendering. *Applied Optics* 56(27):7679-7690, 2017.]

- Collecting public domain High Dynamic Range (HDR) environment maps and shapes and generating random shapes, we can render a representative dataset including ground truth training data.

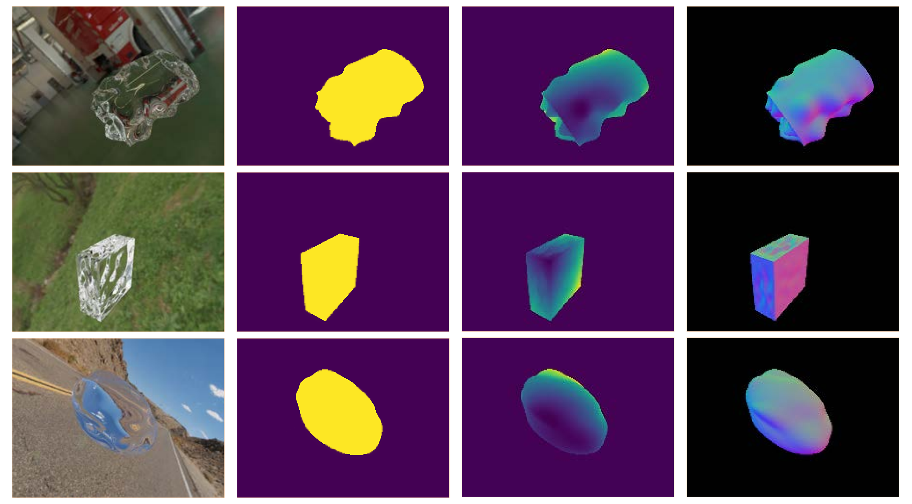
60 HDR Environment Maps

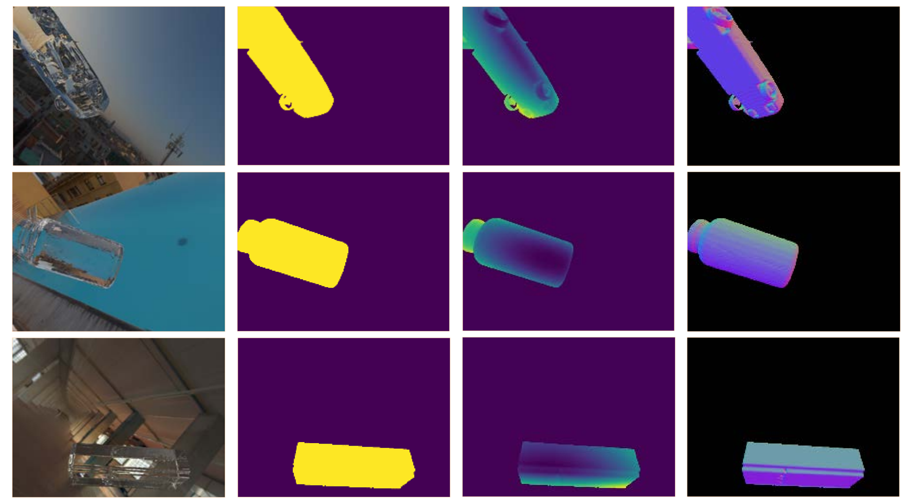


300 Models from ShapeNet

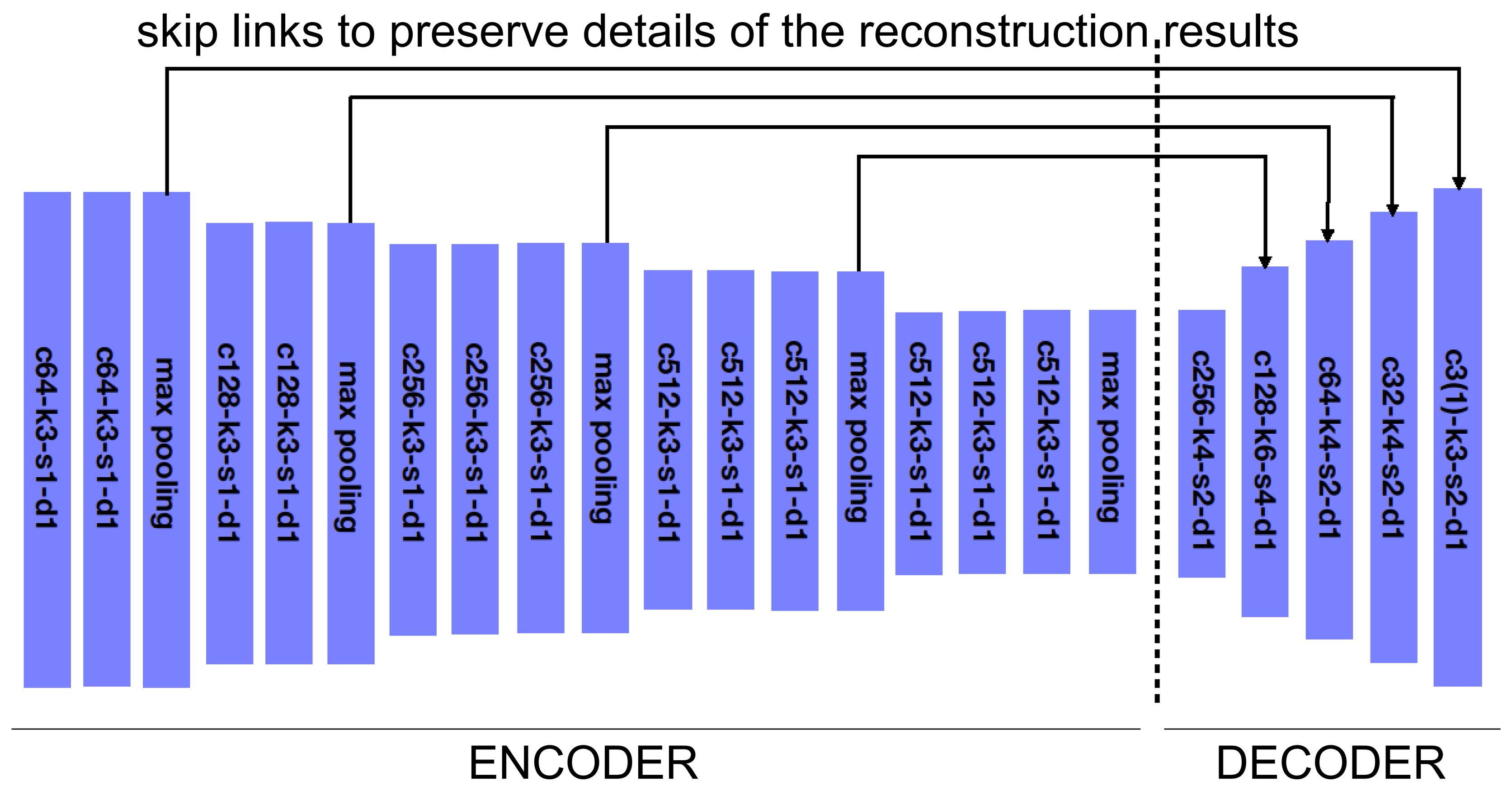
300 Models of generated shapes

- We rendered 80,000 images of the 600 shapes observed from randomly chosen viewpoints. For each RGB-image, we also render segmentation mask, depth, and normal layers as ground truth.





Encoder-Decoder CNN Based on VGG16



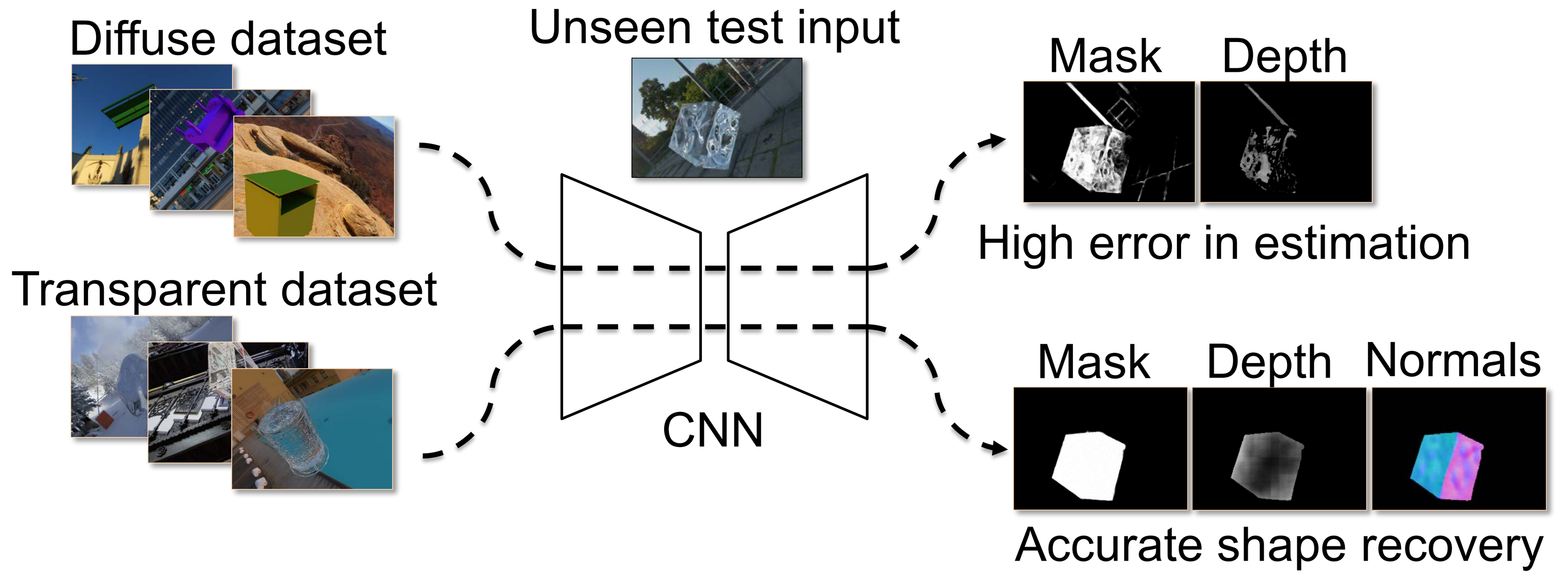
- The output of our network is a segmentation mask, which separates the transparent object from the background, and also a normal map and a depth map. We train a separate encoder-decoder for each task.

Loss Functions

Mask	Depth	Normals
$\frac{1}{\lvert P\rvert}\sum_{p\epsilon P}(M_p-\hat{M}_p)^2$	$\frac{1}{\lvert P_f\rvert}\sum_{p\epsilon P_f}(D_p-\hat{D}_p)^2$	$\frac{-1}{\lvert P_f\rvert}\sum_{p\epsilon P_f} N_p\cdot\hat{N}_p$

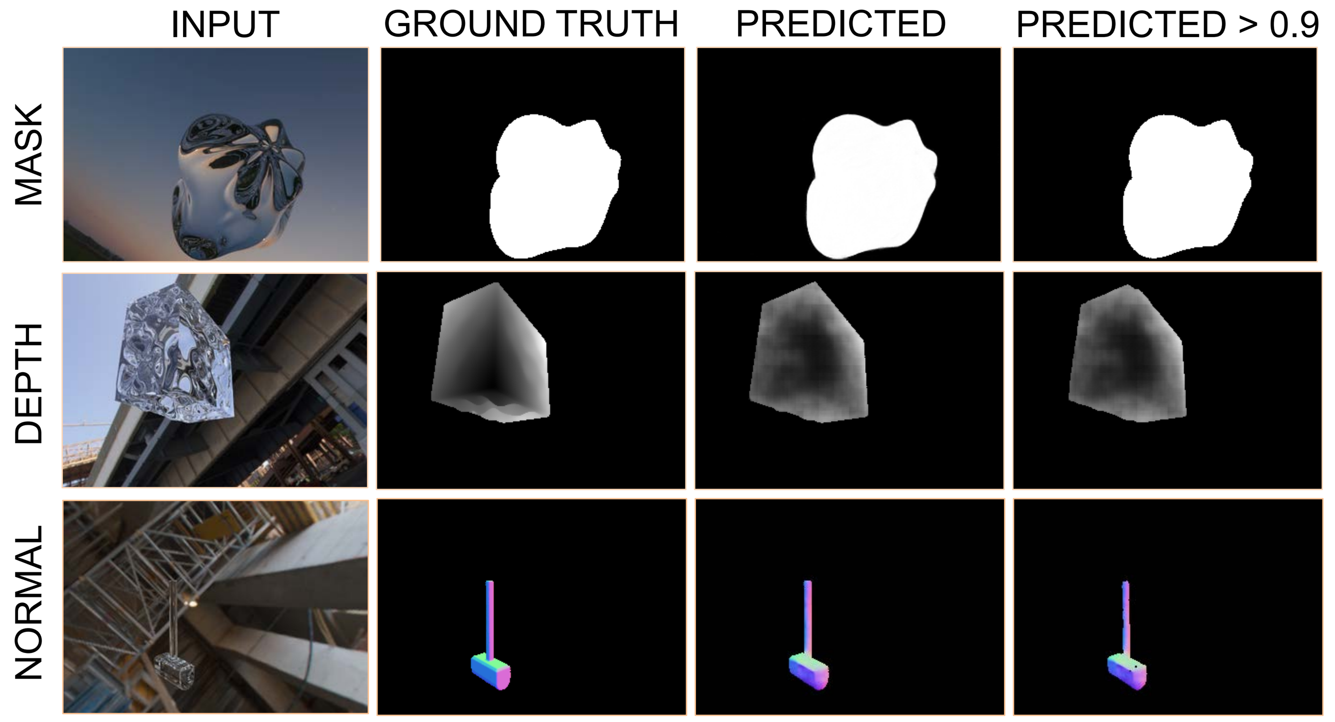
P is the set of all pixels. P_f is the set of masked foreground pixels.

- The following example predictions (when training with diffuse versus transparent shapes) illustrate the need for our transparent dataset, and also the ability of our network to better solve this challenging inverse rendering problem.

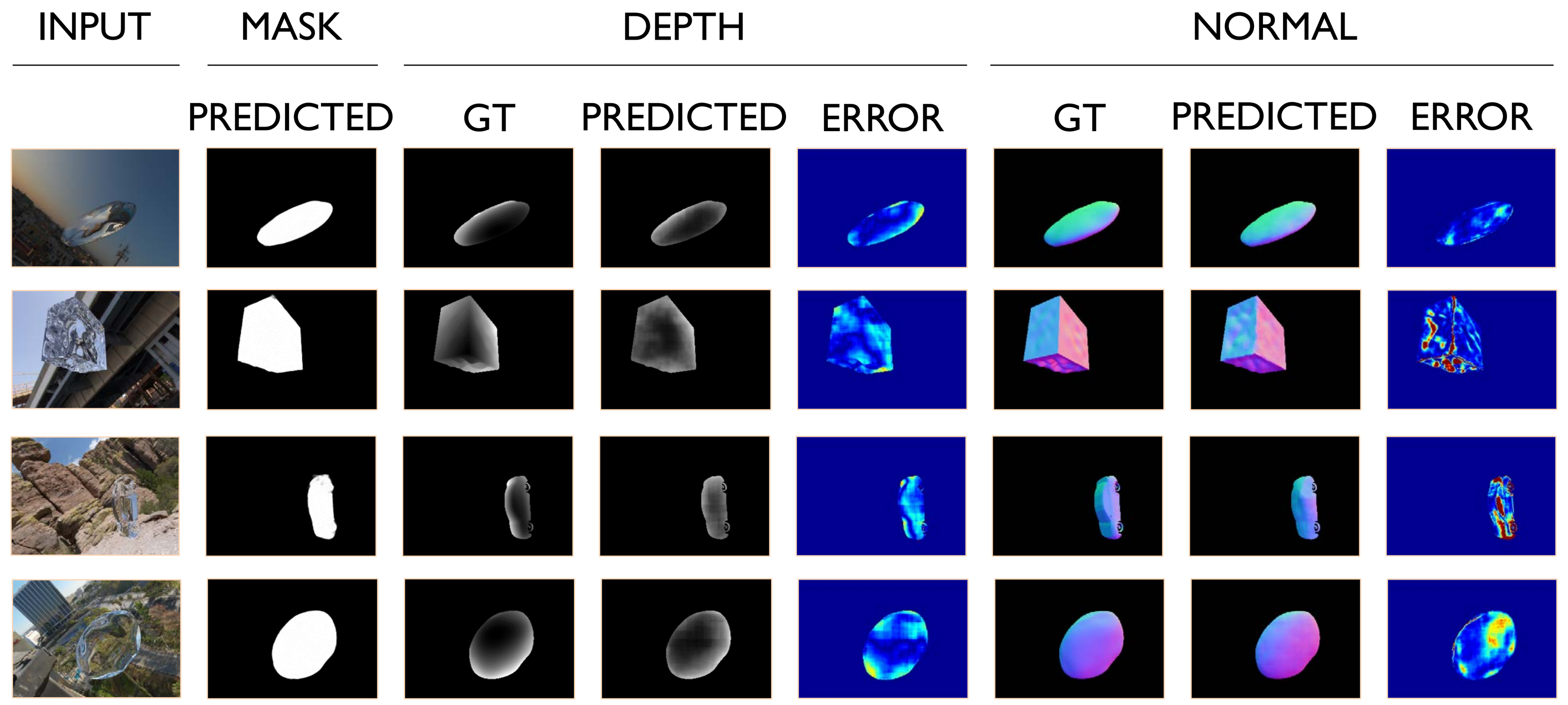


Results

- 0.9 threshold used to binarize the mask.
- Relative depth in [0, 1]: nearest to furthest visible part.
- Depth and normal predictions are masked.



- The error images below display pixelwise loss in the masked regions according to the loss functions, but mapped to a range from 0 (dark blue) to 1 (dark red).



- We used a fixed index of refraction of 1.5 for all shapes in our dataset. Nevertheless, we observe by testing that our network achieves low test errors across a wide range of refractive indices [1.3, 2.0].

- Testing "in the wild" using photographs of glass object.
- Note that our dataset currently does not include thin and hollow shapes.
- Our network was trained entirely with synthetic data, and not fine tuned for specific devices or environments.