DNN-VolVis Interactive Volume Visualization Supported by Deep Neural Network

Fan Hong

Can Liu

Xiaoru Yuan

Slides Made by Haoyu Li

Introduction

Reversed Volume Rendering Application Cases

- Volume data are open-source but transfer function is not published.
- New transfer function design based on current rendering result.

DNN based volume rendering

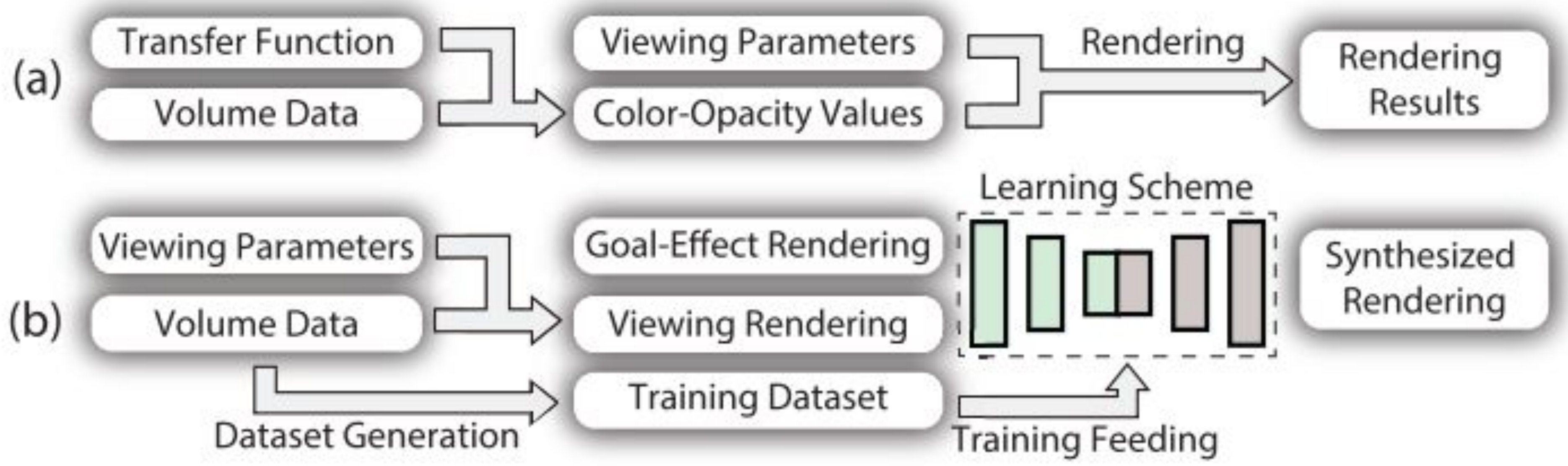

Figure 1: The comparision of traditional volume rendering pipeline and our proposed pipeline. (a) the pipeline of conventional volume rendering, (b) the pipeline of our approach.

Goal

- $I_{new} = G(I_{goal}, V, D)$
- I_{new} : New rendered image
- I_{goal} : Goal effect
- V : Viewing parameters
- D : Volumetric data

Method

Problem Definition

- Image I_{Goal} -> Transfer function C and Viewpoint V_1
- C, New Viewpoint V_2 -> I_{new}

- I_{Goal}, V_2 ---DNN---> I_{new}
- D (Volumetric data), C_0(trivial transfer function), V_2 --> I_2 (Providing detailed image info)

- Thus, I_{Goal}, I_2 ---DNN---> I_{new}

Model Architecture

- Take encoding vectors from Viewing Image and Goal Images instead of random noise vector.
- 3 Discriminators
  - Viewpoint Discriminator
  - Real/Synth Discriminator
  - Transfer Function Discriminator

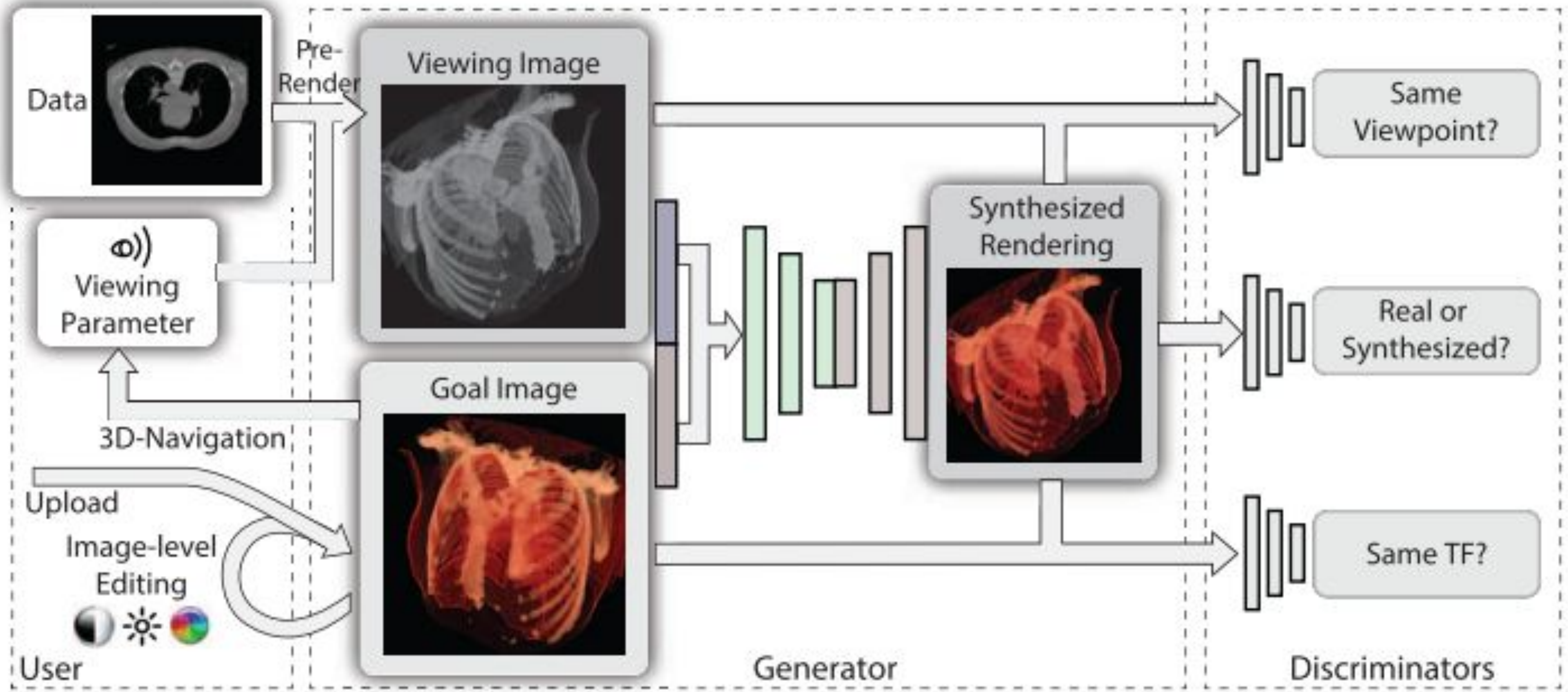

Figure 2: Workflow of our approach including the overall network architecture of our model.

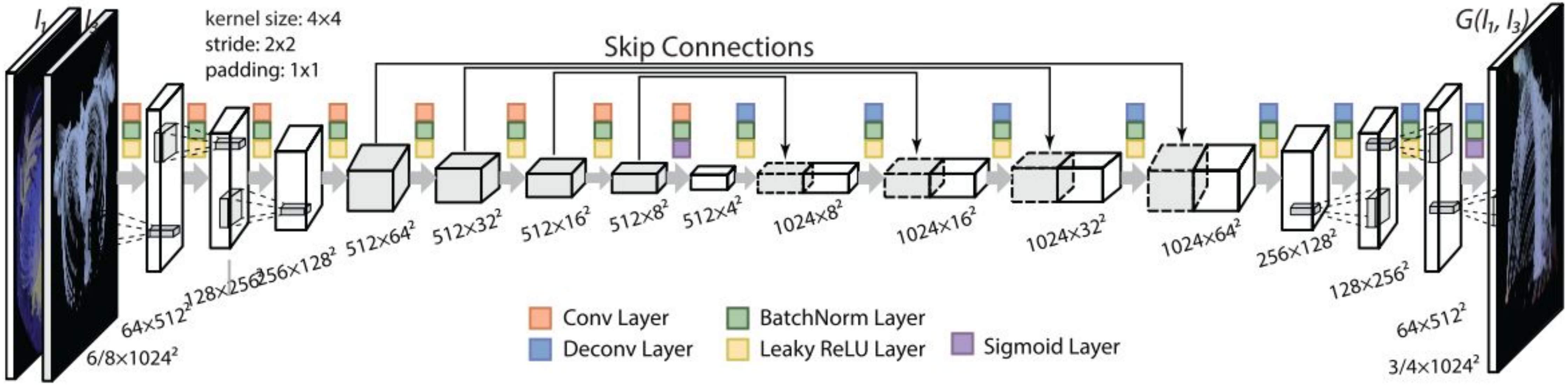

Figure 3: Network architecture of the generator, including encoding and decoding parts. (De)Convolutional layers together with other layers, represented using colored squares, construct the whole network architecture.

U-Net-like skip connection in generator

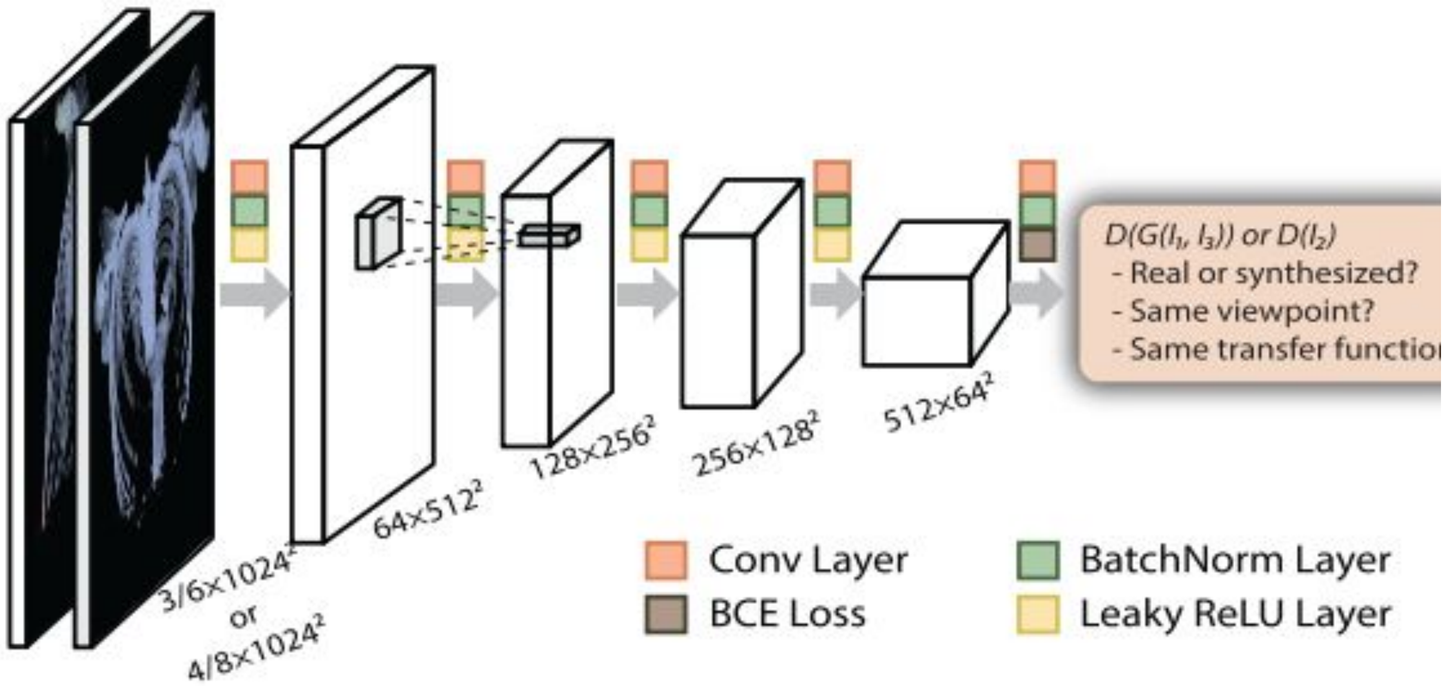

Figure 4: Network architecture of the discriminator. Which ta single image or a couple of images with 3 or 4 channels as input, projects if the image or the image pair to `true` or `false`.

Discriminator (CNN classifier)

Creating Training Samples

- Randomly combine Gaussian Bases which encodes special part in the volume to form Opacity Function.

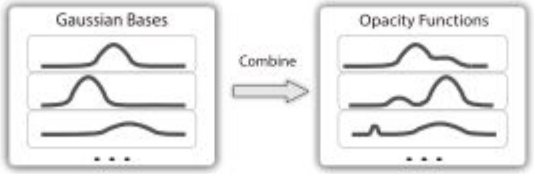

Figure 5: We generate gaussian bases with random mean and deviation; then combine several bases to construct opacity functions

- Construct color function through linear interpolation.
- Randomly sample viewpoints.
- Thus, we have Goal Image, viewing image and new image (as ground truth)

Results

• Goal imageSynthesized image   Ground truth   Difference*5

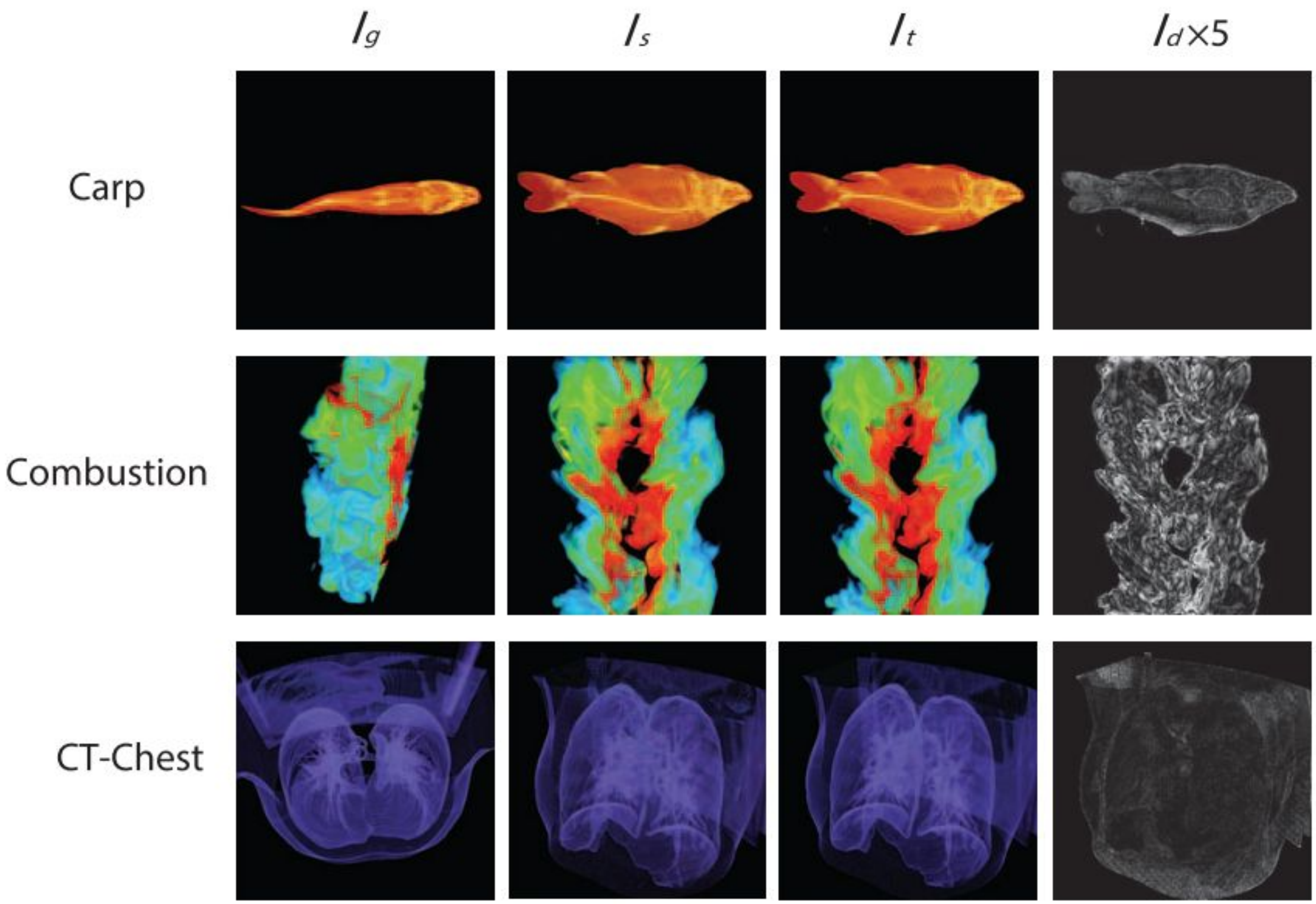

Image Mask Effect

- Alpha channel can mask out background
- When large part of the image is background, gradient vanishing problem occurs.
- Using Alpha channel as input can “cancel” gradient vanishing.

Image Mask Effect

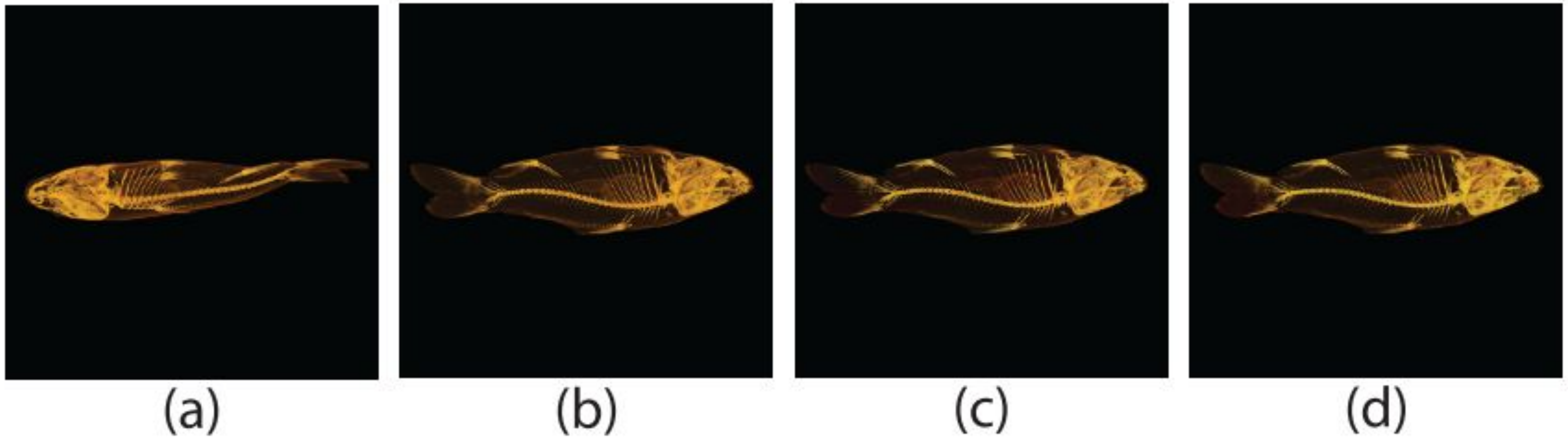

Figure 10: Results of Carp dataset with different configurations. (a) The goal image, (b) the ground truth image, (c) the synthesized images with RGB channels as inputs and the weight for L1 loss λ set to 1000, and (d) the synthesized images with RGBA channels as inputs.

Compare with and w/o skip connection

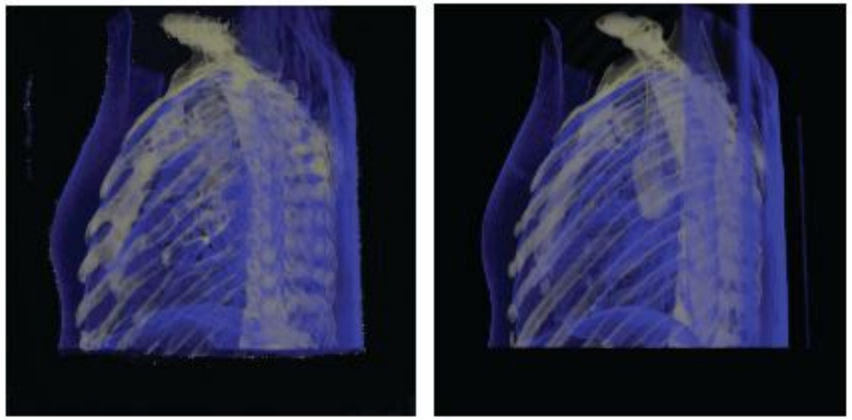

Figure 12: Left without skip layers, right with skip layers

Discussion

Limitation of the method

- If the goal image and the new viewpoint is too far away, we may not have satisfactory result.
- Training data coverage.

Questions

- The method is basically to transfer the goal style to new viewpoint. The paper mentioned they use a trivial transfer function to generate image in new view point but they did not mention how to do it.
- How do you choose this function?
- Will the model still work if the choice is bad?
- If the function choice is not trivial, how do you justify the role of DNN in this paper?