BlockGAN: Learning 3D Object-aware Scene Representations from Unlabelled Images

Thu Nguyen-Phuoc, Christian Richardt, Long Mai, Yong-Liang Yang, Niloy Mitra

Goal

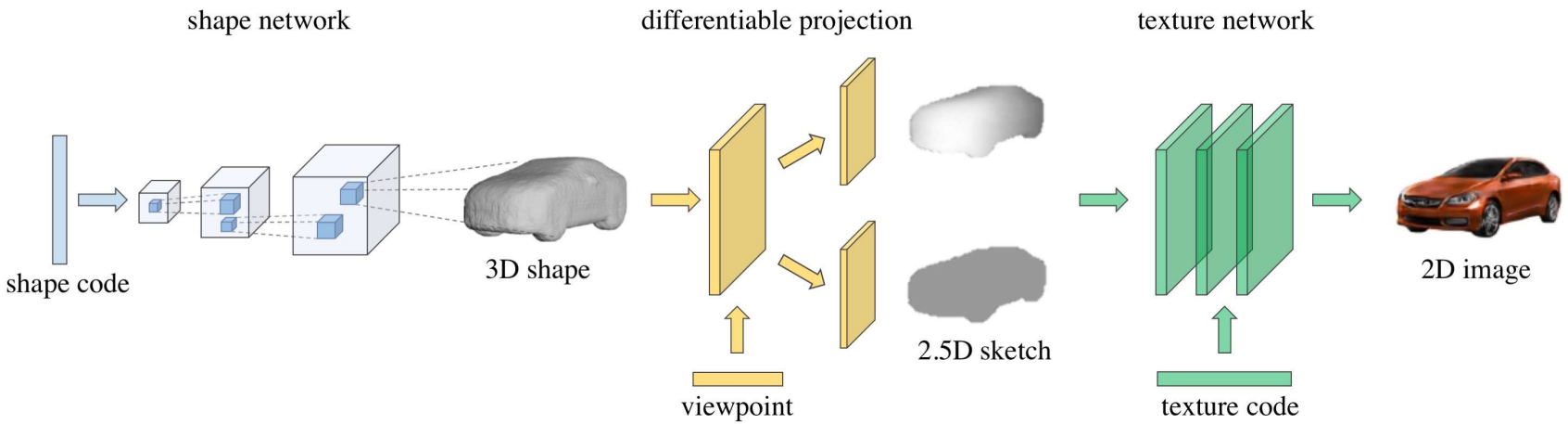
- Learn a network that can capture the distribution of 3D objects
- Requires some disentanglement of shape, texture, view, etc…
- Loooong history of people doing this:
  - 2D based approach
  - 2.5D based approach
  - 3D based approach

Classic 2D based approach

- InfoGAN (Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, Pieter Abbeel) NIPS 2016
  - Information theory based approach
  - Find the latent vectors such that mutual information is maximized

2.5D approach

- Visual Object Networks (Jun-Yan Zhu, Zhoutong Zhang, Chengkai Zhang, Jiajun Wu, Antonio Torralba, Joshua B. Tenenbaum, William T. Freeman) NIPS 2018



(a) samples from WGAN-GP

(b) our 3D, 2.5D, and 2D output

(c) 3D disentanglement

3D approach

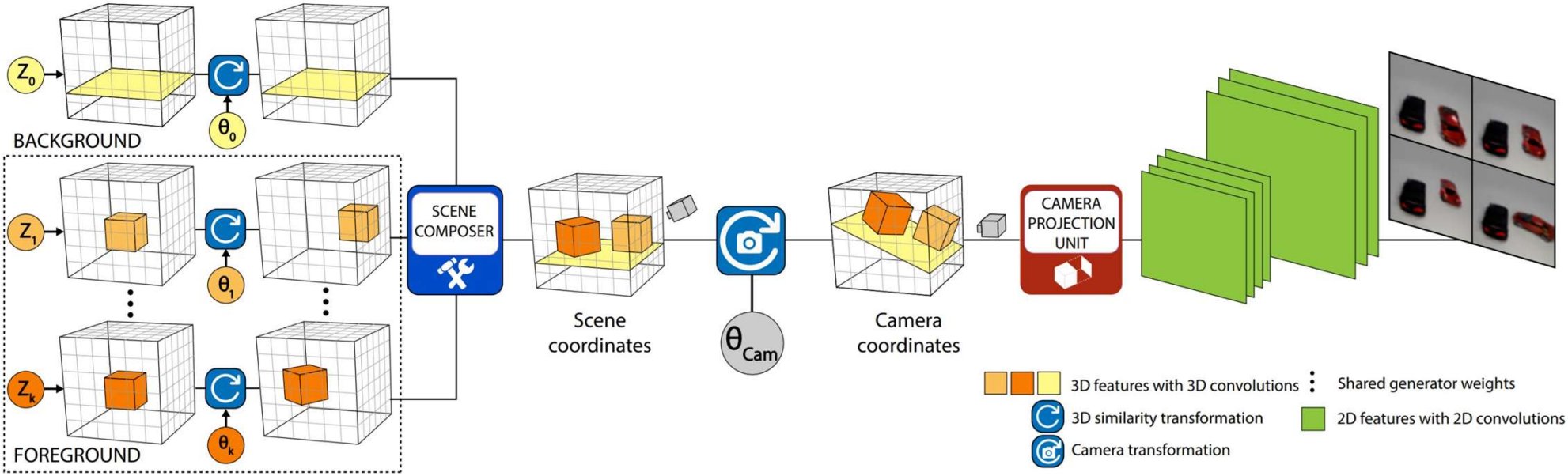
- Scene Representation Networks (Vincent Sitzmann, Michael Zollhöfer, Gordon Wetzstein) NIPS 2019
- This was beaten as of yesterday by NeRF (Neural Radiance Fields paper)

Neural Radiance Fields

BlockGAN approach

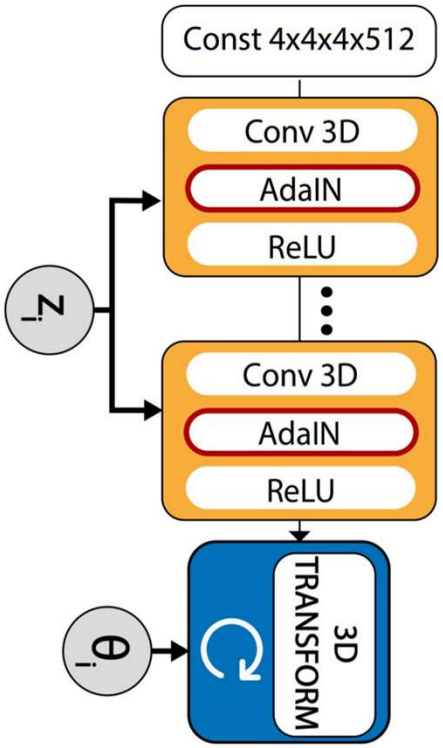
- Previous approaches did not explicitly disentangle objects
  - Either could only model one object, or represented entire scene
- In reality, our world consists of individual physical objects
- Goal would be to utilize this physical prior to learn 3D representations using very minimal assumptions
- Requires: Many images of fixed class, fixed number of objects

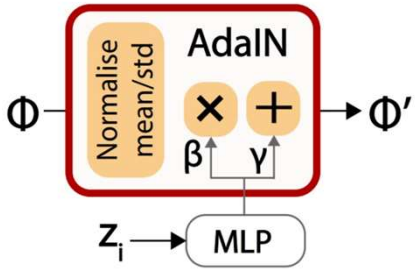
Approach


z_0
θ_0
BACKGROUND
z_1
θ_1
z_k
θ_k
FOREGROUND
SCENE COMPOSER
Scene coordinates
θ_{Cam}
Camera coordinates
CAMERA PROJECTION UNIT
3D features with 3D convolutions
3D similarity transformation
Camera transformation
Shared generator weights
2D features with 2D convolutions

Object generator

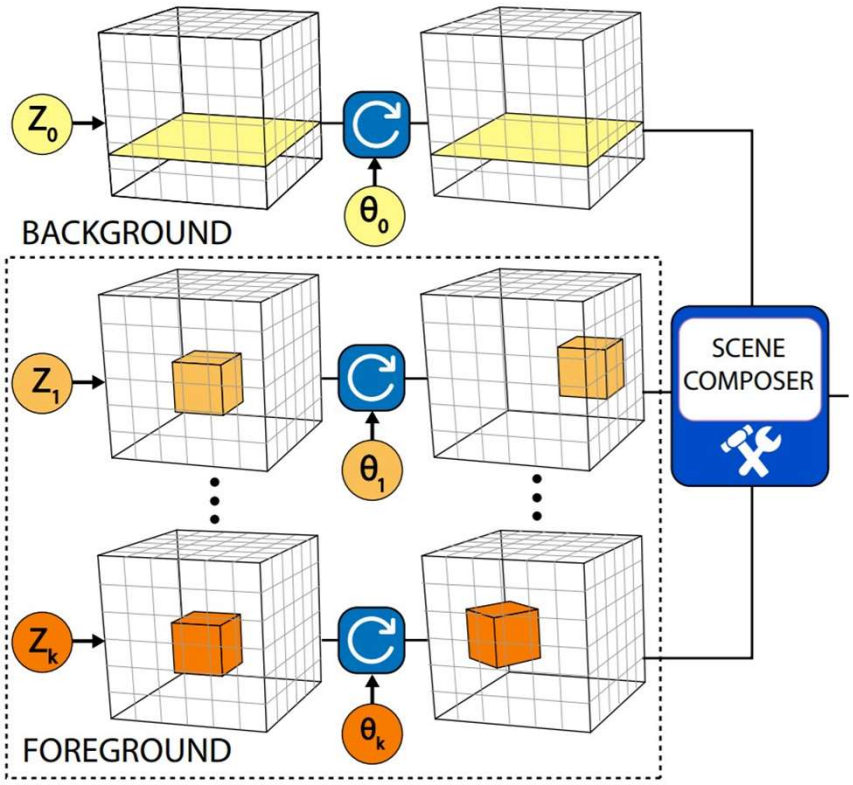
- One network per object
- Identity is encoded in network
- Variance is encoded in the AdaIN layer





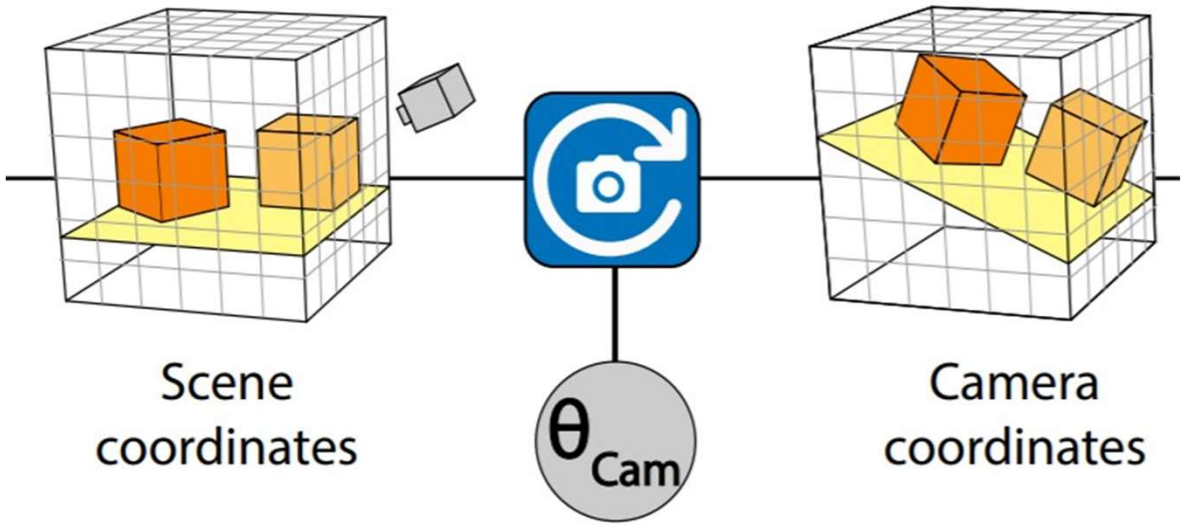
Composition

- Composition function needs to be order-invariant
  - Suppose you add a cube then a sphere
  - The result should be the same as: add a sphere then a cube
- Rules out MLP
- Use avg or max function



3D objects look different depending on view

- Model this explicitly with a view dependent sampling layer

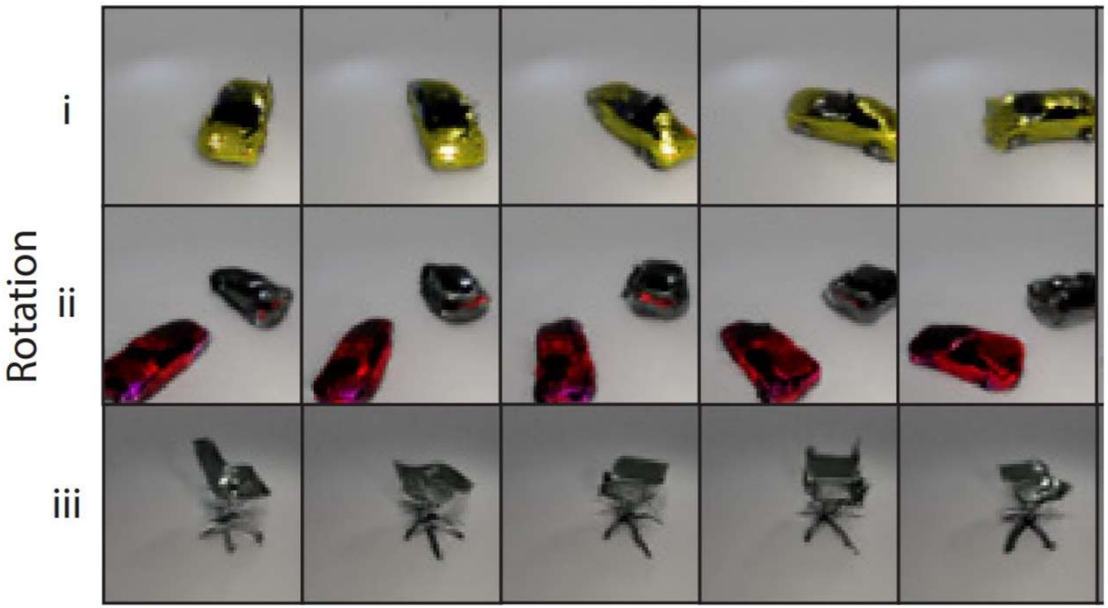


What kind of losses are needed?

- Entire scene must be realistic
  - Use GAN loss
- Statistics of each feature should ideally also match
  - “Style loss”

What it enables

- Manipulation of internal 3D representations to change image



Works even on real images

Why we care

- Humans acquire object permeance at a very early age
- We are able to easily imagine manipulating 3D objects with ease
- Clearly humans can isolate shape/texture/lighting
- Disentangled 3D representations enables us to perform “surgery” on internal network features and directly visualize the output