COMPUTER VISION

Project Final-Evaluation
29th April, 2021

Namratha Gopalabhatla (20171017)
Sagrika Nagar (20171204)
Nishant Sachdeva (2018111040)
Sumanth Balaji (2018114002)

Problem Statement:

Explore the use of implicit fields for learning generative models of shapes and introduce an implicit field decoder for shape generation, aimed at improving the final visual quality of these generated models.

Steps in Implementation:

1. Understanding the Paper
2. Data Gathering and Preparation
3. Implementation of Auto-encoder
4. Implementation if IM-NET Decoder
5. Implementation of GAN
6. Results
7. Challenges and Possible improvement areas

Problem

- Typical state-of-the-art architectures for 3D generative shape modelling using standard CNNs and GANs are poor in terms of visual quality .
- They exhibit discontinuous and overly smoothed surfaces, provide low resolution outputs and are susceptible to irregularities in training data.
- CNNs learn voxel distributions over a volume, rather than the shape boundary itself

Solution

- Implicit Fields used in Decoder feeds the point coordinates along with shape feature vector to determine whether a certain point lies on the inside or the outside, relative to the shape .
- The method allows to learn shape boundaries and output at multiple resolutions, irrespective of the resolution of the training data.
- This shape aware network produces shapes of higher visual quality on interpolation through latent GANs

Definitions

- **VOXEL**
  A pixel is a 2-dimensional Raster graphic, having the values of width and length, with colour placed inside the coordinate. A voxel is a raster graphic on a 3-dimensional grid, with the values of length, width and depth. It also contains multiple scalar values such as opacity, color and density.
- **IMPLICIT SURFACE**
  Implicit surfaces are actually explicit volumes, as in, they explicitly define what is the inside and outside of the object. Consider the function which takes a 3D point (x,y,z) as input and returns a single value. This value tells us explicitly if we are outside or inside the volume. This function is also sometimes loosely called an Implicit Field.

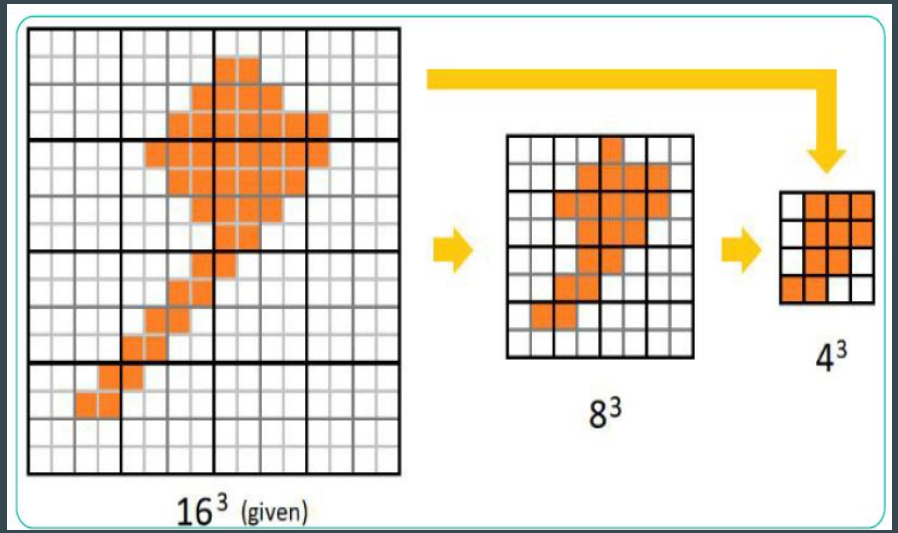
Definitions

- **ISO-SURFACE**
  An isosurface is a three-dimensional surface that represents points of a constant value within a volume of space. Thus for our purpose, the object we need to construct has a zero ISOsurface. A mesh surface can be reconstructed by finding the zero isosurface of the implicit field.

Data Preparation

We need a point value cloud for the training of our implicit decoder.

For 3D shapes, to get voxel models in different resolutions (16x16x16 to 128x128x128), we sample points on each resolution in order to train the model progressively.



Data Preparation

However, a naive sampling would imply taking the center of each voxel and thus produce ***n*** points in each dimension i.e., ***n x n x n*** points.

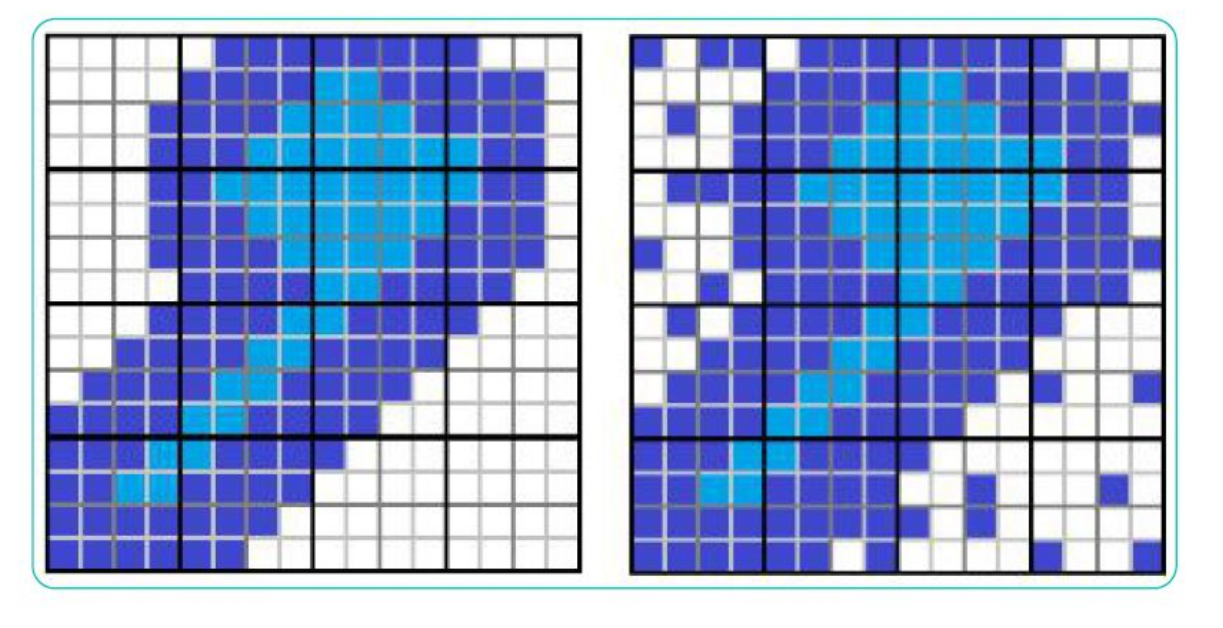
We aim to get n x n points and thus sample more points closer to the shape surfaces and neglect points far away.

To compensate for a density change, we assigned weights of all sampled points to 1, because we want the model to pay more attention to the surface and allow small errors in the void area.

Voxel grid resolution	16^3	32^3	64^3	128^3
Number of points	16^3	$16^3 \times 2$	32^3	$32^3 \times 4$

Data Preparation

Sample points which are within 3 voxel (in all x, y, z directions) from shape boundaries are taken. If the number of sampled points does not exceed the limit, randomly sample more points up to a limit.



Dataset Sample

Airplane dataset image sample at different resolutions and it is visualized at 3 axes:

RESOLUTION 16:



Dataset Sample

RESOLUTION 32:

Dataset Sample

RESOLUTION 64:



IM-NET

Implicit Field Decoder: IM-NET

- A simple and generic implicit field decoder to learn shape boundaries.
- The network takes as input a feature vector extracted by a shape encoder, as well as a 3D or 2D point coordinate, and it returns a value indicating the inside/outside status of the point relative to the shape.
- The loss function used to train the complete model is a weighted mean squared error between the ground truth labels and the labels predicted by the model for each sampled point of the target shape.

$$\mathcal{L}(\theta) = \frac{\sum_{p \in S} |f_\theta(p) - \mathcal{F}(p)|^2 \cdot w_p}{\sum_{p \in S} w_p}$$

IM-NET Structure

Layer	Input Shape	Activation
f.code + coordinates	(128+3)	-
fully-connected	(131)	LReLU
fully-connected	(1024+131)	LReLU
fully-connected	(1024+131)	LReLU
fully-connected	(1024+131)	LReLU
fully-connected	(512+131)	LReLU
fully-connected	(256+131)	LReLU
fully-connected	(128)	Clip Function

IM-NET

- In a typical application setup, our decoder, would follow an encoder which outputs the shape feature vectors and then return an implicit field to define an output shape.
- The implicit field decoder, can be embedded into different shape analysis and synthesis frameworks to support various applications.

Features of IM-NET

- **Resolution**
  Decoder output can be samples at any resolution and thus not hindered by the resolution of the training samples. In our case, we sampled the output shapes at 256x256x256
- **Properties Learned**
  Since point coordinates are also concatenated with 128-feature vector, the network learns the inside/outside status of any point relative to a shape. In contrast, a CNN network predicts possibility of each pixel to be on/off relative to extent of bounding volume of a shape.

IM-AE

Encoder

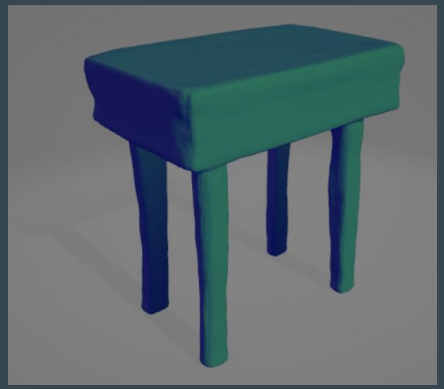
- For auto-encoding 3D shapes, we used a 3D CNN as encoder to extract 128-dimensional features from 64x64x64 voxel models.
- We use IM-NET decoder instead of a CNN decoder and compare the results.
- CNN-AE predicts the possibility for each specific pixel to be on or off, while IM-AE learns the field of the shape.

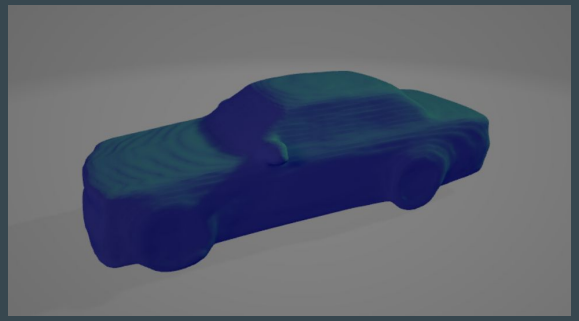
Encoder Structure for IM-AE

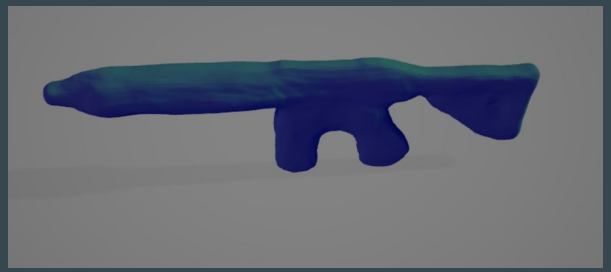
Layer	Kernel Size	Stride	Activation
Input voxels	-	-	-
conv3d	(4,4,4)	(2,2,2)	BatchNorm LReLU
conv3d	(4,4,4)	(2,2,2)	BN LReLU
conv3d	(4,4,4)	(2,2,2)	BN LReLU
conv3d	(4,4,4)	(2,2,2)	BN LReLU
conv3d	(4,4,4)	-	Sigmoid

Results for IM-AE

Results for IM-AE







Results

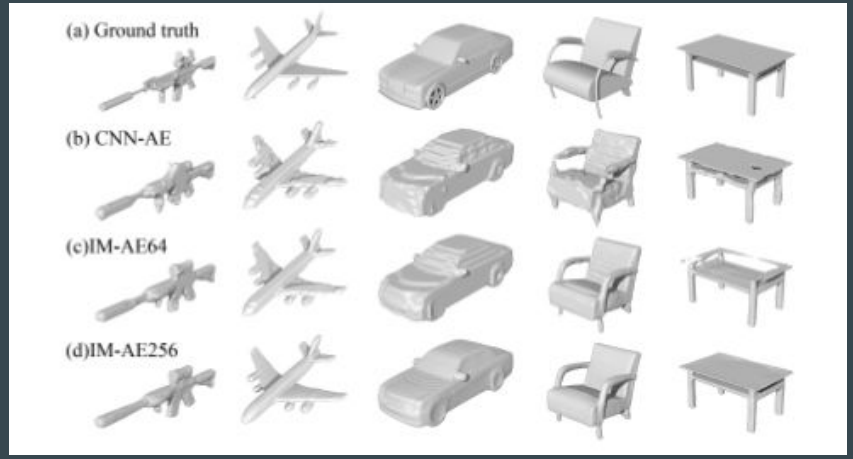
Metrics Used:

- Chamfer Distance (CD)
- Mean Squared Error (MSE)
- Intersection over Union (IoU)
- Light Field Descriptor (LFD)

	Plane	**Car**	**Chair**	**Rifle**	**Table**
CNN-MSE IM-MSE	1.47 3.17	4.37 5.01	7.76 9.32	1.62 2.03	5.80 9.46
CNN-IoU IM-IoU	86.07 80.01	90.73 89.52	74.22 64.72	78.73 71.92	84.67 73.11
CNN-CD IM-CD	3.51 4.25	5.31 5.43	7.34 9.12	3.48 3.81	7.45 11.75
CNN-LFD IM-LFD	3375 3224	1323 1150	2555 2461	3515 3613	1824 2230

*CNN-AE Results were taken from the paper

Results



Although CNN-AE beats IM-AE in nearly all five categories in terms of MSE, IOU, and CD, visual examination clearly reveals that IM-AE produces better results. This validates that LFD is a better visual similarity metric for 3D shapes.

IM-GAN

IM-GAN

- IM-NET learns shape boundaries while CNN learns voxel distributions over a volume. Thus, IM- NET is particularly useful in case of GANs as shape evolution is a direct result of changing the assignments of point coordinates to their inside/outside status.
- We use latent GANs on feature vectors learned by IM-AE. This way, the pretrained encoder would serve as a means for dimensionality reduction, and the latent-GAN was trained on high-level features of the original shapes.
- We used two hidden fully connected layers for both the generator and the discriminator, and the Wasserstein GAN loss with gradient penalty.

Latent GAN Structure

Generator:

Layer	Activation Function	Output Shape
latent vector	-	(128)
fully-connected	LReLU	(2048)
fully-connected	LReLU	(2048)
fully-connected	Sigmoid	(128)

Discriminator:

Layer	Activation Function	Output Shape
feature code	-	(128)
fully-connected	LReLU	(2048)
fully-connected	LReLU	(2048)
fully-connected	-	(1)

Training Configuration

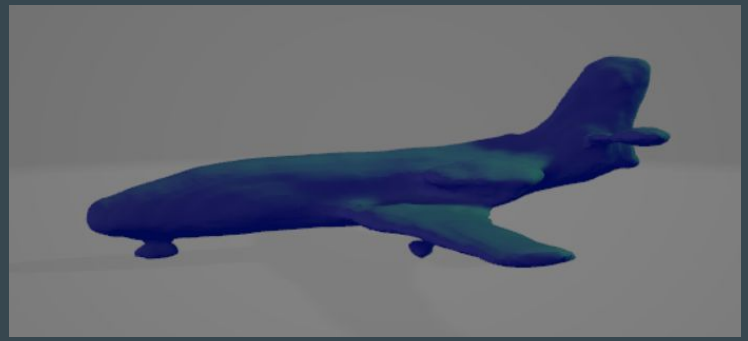
- The networks were implemented with PyTorch and using Adam optimizer (learning_rate=5e-5, beta1=0.5, beta2=0.999, epsilon=1e-8).
- For leaky ReLU, alpha=0.02.
- For batch normalization, decay=0.999, epsilon=1e-5.
- The training batch size is: 32 for 3D CNN-based models, 1 for implicit-decoder-based models, 50 for latent-GANs.
- Notice that for implicit-decoder-based models, the batch size is one shape, the actual batch size for implicit decoder varies according to the resolution of the training data.

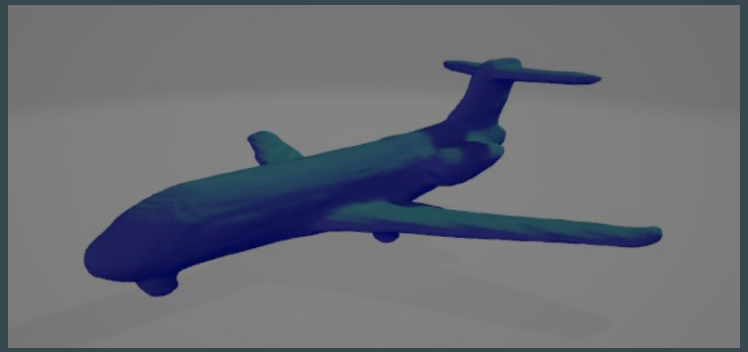
Results

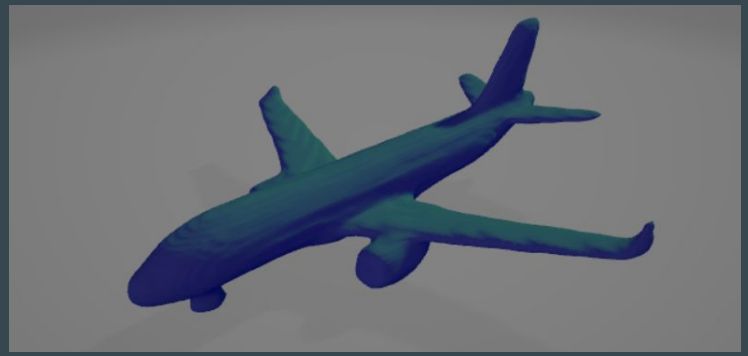





Results







Evaluation Metrics

- Coverage Score (COV-LFD)

Type of GAN	Plane	Car	Chair	Rifle	Table	Average
CNN-GAN	69.22	73.00	77.33	61.26	83.73	72.99
IM-GAN	70.35	69.23	75.44	66.10	86.43	73.51

- Minimum Matching Distance (MMD-LFD)

Type of GAN	Plane	Car	Chair	Rifle	Table	Average
CNN-GAN	3745	1288	3012	3819	2954	2892
IM-GAN	3690	1289	2894	3765	2528	2833

Results

Metrics Used:

- Coverage Score (COV-LFD)
- Minimum Matching Distance (MMD-LFD)

Overall, IM-GAN performs better on both COV-LFD and MMD-LFD. More importantly, IM-GAN generates shapes with better visual quality compared to other methods, in particular, with smoother and more coherent surfaces.

Issues, Improvements and Applications

- We can use the ResNet encoder to obtain 128-D features from 128^2 images. By only training the ResNet encoder (to minimize the mean squared loss between the predicted feature vectors and the ground truth), we can perform better than training the image-to-shape translator directly, since one shape can have many different views (possibly leading to ambiguity)
- A key merit of our implicit encoder is the inclusion of point coordinates as part of the input feature, but this comes at the cost of longer training time, since the decoder needs to be applied on each point in the training set.

Issues, Improvements and Applications

- When retrieving generated shapes, CNN only needs one shot to obtain the voxel model, while our method needs to pass every point in the voxel grid to the network to obtain its value, therefore the time required to generate a sample depends on the sampling resolution.
- Dissimilar Shapes: Cannot ensure a meaningful morph between highly dissimilar shapes ie, from different categories.
- Low Frequency Errors: Like global thinning and thickening of meshes.

Thank You