https://arxiv.org/pdf/2210.00379.pdf

Neural radiance field models are novel view synthesis methods which use volume rendering with implicit neural scene representations (MLPs) to learn geometry and lighting of a 3D scene

- Self- supervised: trained using only multi view images of a scene, require only image and poses to learn a scene (no depth supervision)
    - Poses can be estimated using SfM even
- Photo realistic results
- Represents 3D scene as radiance field by NN
    - Radiance field describes color and volume density for every point and viewing direction in scene

$F(\mathbf{x}, \theta, \phi) \rightarrow (\mathbf{c}, \sigma),$ approximated by NN

x = (x, y, z) in scene coordinates

(θ, φ) = azimuthal and polar viewing angle, also d in 3D cartesian unit vector

c = (r, g, b) color

- Allowed to be dependent on both viewing direction and coordinate

Sigma is volume density

- Constrained to be viewing direction independent

Architecture

- First stage takes input x and outputs sigma and a high dimensional feature vector
- Second stage, feature vector is concatenated with viewing direction d and passed to an additional MLP which outputs c

Three scene representations: implicit, hybrid, explicit

- Baseline NeRF, density and color fields are fully represented by MLPs (explicit)

Novel view synthesis steps

- For each pixel in the image being synthesized, send camera rays through the scene and generate a set of sampling points (see (a) in Fig. 1).
- For each sampling point, use the viewing direction and sampling location to compute local color and density using the NeRF MLP(s) (as shown in (b) in Fig. 1).
- Use volume rendering to produce the image from these colors and densities (see (c) in Fig. 1).

$$C(\mathbf{r}) = \int_{t_1}^{t_2} T(t) \cdot \sigma(\mathbf{r}(t)) \cdot \mathbf{c}(\mathbf{r}(t), \mathbf{d}) \cdot dt,$$

Color C(r) of any camera ray r(t) = o + td (camera position o and viewing direction d)

sigma(r(t)) volume density at point r(t) along camera ray

c(r(t), d) color at point r(t) along the camera ray viewing direction d

Dt is differential distance traveled by ray

T(t) is accumulated transmittance, probability that ray travels from t1 to t without being intercepted

$$T(t) = \exp(-\int_{t_1}^{t} \sigma(\mathbf{r}(u)) \cdot du).$$

Novel views are rendered by tracing camera rays C(r) through each pixel of the to-be-synthesized images

- Non- deterministic stratified sampling where ray was divided into N equally spaced bins and sample was uniformly drawn from each bin

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} \alpha_i T_i \mathbf{c}_i, \text{ where } T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j).$$

Delta is the distance from sample i to sample i+ 1
(sigma, c) are the density and color evaluated along the sample point
Alpha is the transparency/ opacity given by 1- exp(sigma, delta)

$$d(\mathbf{r}) = \int_{t_1}^{t_2} T(t) \cdot \sigma(\mathbf{r}(t)) \cdot t \cdot dt.$$

e approximated analogously to equati
g equation (2) and (3)

$$\hat{D}(\mathbf{r}) = \sum_{i=1}^{N} \alpha_i t_i T_i. \qquad L = \sum_{r \in R} ||\hat{C}(\mathbf{r}) - C_{gt}(\mathbf{r})||_2^2$$

NeRF can also use positional encoding to improve fine detail reconstruction

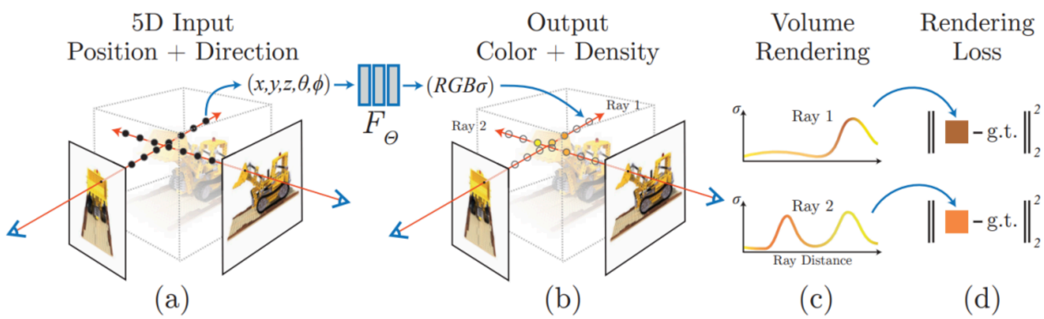


Fig. 1. The NeRF volume rendering and training process. Image sourced from [1]. (a) illustrates the selection of sampling points for individual pixels in a to-be-synthesized image. (b) illustrates the generation of densities and colors at the sampling points using NeRF MLP(s). (c) and (d) illustrate the generation of individual pixel color(s) using in-scene colors and densities along the associated camera ray(s) via volume rendering, and the comparison to ground truth pixel color(s), respectively.

Datasets

- A lot use blender to generate the datasets
- COLMAP to compute poses (SfM)
- Conventional
    - Realistic synthetic 360
    - LLFF dataset

    - DTU dataset
    - ScanNet
    - ShapeNet
- Building scale
    - Tanks and temples
    - Matterport- 3D
    - Replica
- Large scale urban
    - KITTI
    - Waymo Open
- Human Avatar/ Face
    - Nerfies/ HyperNeRf
    - CMU Panoptic
    - NeuMan

Quality metrics (quality of individual images with or without ground truth images):

- PSNR: peak signal to noise ratio (no reference)
- SSIM: structural similarity index measure (full reference)
- LPIPS: learned perceptual image patch similarity (full reference)

Improvements in quality of synthesized views and learned geometry

View synthesis

- Mip- NERF: cone tracing instead of ray tracing of standard NeRF volume rendering (cone was Gaussian from camera center)
- Mip- NERF 360: proposal MLP, supervised using NERF MLP but not images
    - Proposal MLP only predicts volume density and not color and used to find appropriate sampling intervals
    - Novel scene parameterization for Gaussians
    - New regularization to prevent floating geometric artifacts and background collapse
- Ref- NERF: reflection of viewing direction about the local normal vector
    - Modified density MLP into a directionless MLP which outputs not only density and input feature vector but also BRDF
- RegNERF: aimed to solve problem of NeRF training with sparse input views
    - Just used additional depth and color regularization unlike other methods which used image features from pretrained networks as a prior
- Ray Prior NeRF: view extrapolation instead of interpolation
    - Random ray casting: given a training ray hitting a surface point, a backward ray was cast starting from v towards a new origin
    - Ray Atlas: extracting rough 3D mesh from pretrained nerf and mapping training ray directions onto 3D vertices

Depth supervision and point cloud

- DS Nerf: depth supervision with sparse point clouds from training images
    - KL divergence between ray and noisy depth distribution
- NerfingMVS: depth map prior to supervise volume sampling by only allowing sampling points at appropriate depth
- PointNeRF: feature point clouds as an intermediate step to volume rendering
    - Pretrained 3D CNN to generate depth and surface probability from a cost volume
    - Pretrained 2D CNN to extract image features from training views
    - PointNet like network to regress local density and color for volume rendering

Other geometry improvements

- SNeS: improved geometry by learning symmetries through soft symmetry constraints on geometry and material
- S3 NERF: used shadow and shading to infer scene geometry and achieved single image NeRF training for geometry recovery
    - UNISURF based occupancy field 3D representation instead of density, modified physics based rendering equation, occupancy based shadow calculation

Improvements to training and inference speed

Originally they used hierarchical rendering to improve computation efficiency

- Naive rendering would require densely evaluating MLPs at all query points along each camera ray
- Course network to pick sampling points for fine network which prevented dense sampling at fine scale

Train, precompute, store NeRF MLP evaluation results into more accessible data structures, “baked models”

- Improve inference speed
- SNeRG: sparse neural voxel grid, stored a precomputed NeRF (diffuse color, density, and feature vectors) on a sparse voxel grid
    - During evaluation, MLP was used to produce specular color and alpha
- PlenOctree: Built octree of precomputed spherical harmonic coefficients of MLP’s colors
    - Trained a spherical harmonic NeRF(but method can also be done to normal nerf) which predicted spherical harmonic coefficients of color function instead of directly color function
    - Scene first voxelized with low transmissivity voxels eliminated
        - Can be further optimized using training images which is fast
- FastNeRF: factorized color function into inner product of output of direction position-dependent MLP and output of a direction- dependent MLP
    - Cache color and density evaluations
    - Early ray termination and empty space skipping also
- KiloNeRF: separated scene into thousands of cells and trained independent MLPs for color and density predictions on each cell
    - Independent MLPs trained using knowledge distillation from a large pre- trained teacher MLP

- Fourier Plenoctree: built for human silhouette rendering because it used domain specific, Shape- From- Silhouette
    - Coarse visual hull using sparse views predicted from generalized NeRF and shape from silhouette
    - Then color and density densely sampled and stored in coarse Plenoctree
    - Fourier transformed
- MobileNERF: nerf like based on polygonal mesh with color, feature, opacity MLPs attached to each mesh point
    - During rendering, mesh with associated features and opacities are rasterized based on viewing direction
- EfficientNeRF: based on PlenOctree
    - Momentum density voxel grid to store predicted density
    - Pivot system: only points near pivot points are considered during fine sampling
- NeLF: distill pretrained NeRF into surface light fields

Non- baked models
- Sometimes learn separate scene features from learned MLP parameters, hybrid scene representations
    - Smaller MLPs such as voxel grid instead of radiance field
    - Improve training and inference speed at the cost of memory
- Ray termination (prevent further sampling points when accumulated transmittance approaches zero), empty space skipping, hierarchical sampling (coarse + fine MLPs)
- JaxNeRF: re implementation of original NeRF in JAX which was slightly faster and more suited for distributed computing
- NeRF adjacent methods that use learned voxel, tree features instead and skip over MLPs and performed volume rendering directly on the learned features
- NSVF: Neural sparse voxel fields, voxel- based NeRF model that models scene as a set of radiance fields bounded by voxels
    - Features obtained by interpolating learnable features stored at voxel vertices
- AutoInt: approximates volume rendering by separating volume rendering equation piecewise
- Deterministic integration for volume rendering: similar to NSVF, deterministic ray sampling on voxel grid which produced an integrated feature for each ray interval (decoded by MLP for density and color)
- Instant neural graphics primitives: learned parametric multi resolution hash encoding that was trained simultaneously with NeRF model MLPs

Few shot/ sparse training view NeRF
Baseline NeRF requires dense multi- view images with known camera poses for each scene
- Failure case of training views not being varied enough: too few samples, samples not varied enough in pose
- Leverage pretrained image feature extraction networks (CNN) to lower training samples required, “image feature conditioning”
- Depth/ 3D geometry supervision

- PixelNeRF: pretrained layers of CNN to extract image features
    - GRF (general radiance field): took a similar approach but this operated in canonical space while pixelNERF was in view space
- MVSNeRF: 2D image features mapped to 3D voxelized cost volume using plane sweeping and a variance based cost
    - 3D CNN extracts 3D neural encoding to generate per point latent codes
    - Joint optimization of 3D feature volume and NeRF MLP
- DietNeRF: semantic consistency loss based on image features in addition to standard photometric loss
- NeuRay: cost volumes from MVS
    - Alpha based sampling strategy: only sampling around points with high hitting probability
- GeoNERF: 2D image features from pretrained feature pyramid + 3D cost volume
    - For each query point along ray: one view independent and multiple view dependent tokens were extracted which were refined using Transformer
    - View- independent tokens refined through an autoencoder
    - Networks can all be pretrained
- LOLNeRF: single shot view synthesis of human faces, generative latent optimization instead of adversarial training
- NeRFusion: Also 3D cost volume from 2D image features
    - GRU to fuse these local feature volumes into a global feature volume
- AutoRF: novel view synthesis of objects without background
    - 3D object detection algorithm to extract 3D bounding boxes and object masks
    - Normalizing Object Coordinate spaces for per- object volume rendering
    - Additional occupancy loss per object
- SinNeRF: integrate multiple techniques for single image NeRF
    - Image warping, known camera parameters for depth supervision of unseen views
    - Adversarial training with CNN to provide patch wise texture
    - Pretrained ViT to extract global image features
- GeoAug: solve few shot using data augmentation by rendering (with warping) new training images with noisy camera poses

Generative and Conditional Models

Generative NeRF models generate 3D geometry conditioned on texts, images, latent codes + scene editing

2D generative models to create images of the scene which are then used to train NeRF

- Key challenge of controlling the pose and keeping the subject invariant

- NeRF- VAE: generalized well to OOD scenes and removed need to train on each scene from scratch
    - NeRF renderer conditioned on latent code using iterative amortized inference and a resnet based encoder
    - Attention based scene function

    - Outperform baseline NeRF on low view scenes

GAN methods

- GRAF: first NeRF model trained adversarially
    - NeRF based generator conditioned on latent appearance code and shape code
    - Shape code for conditioning scene density + embedded position were inputted into direction independent MLP
    - Appearance code for conditioning scene radiance + direction dependent MLP
    - Images generated via volume sampling and then compared using discriminator CNN
    - Assigned each object in the scene an MLP
- Pi- GAN: SIREN based NeRF volumetric renderer
- EG3D: novel hybrid tri- plane representation with features stored on three axis aligned planes and a small decoder MLP for neural rendering in a GAN framework
    - GAN uses a pose-conditioned StyleGAN2 feature map generator, NeRF rendering module converting tri-plane features into low resolution images, super resolution module
    - Super resolution then fed into StyleGAN2 discriminator
- StyleNeRF: use NeRF to bring 3D awareness to StyleGAN image synthesis
    - Style code conditioned NeRF
- Pix2NeRF: adversarially trained NeRF which could generated NeRF rendered images given randomly sampled latent codes and poses
    - Built from pi- gan

Jointly Optimized Latent Models

Jointly optimize latent code using Generative Latent Optimization with the scene model

- Not generative but use latent codes for various changeable aspects of the scene
- Can be thought of us as a Discriminator-less GAN
- Edit-NeRF: scene editing using image conditioning from user input

CLIP- NeRF: extract from user input text or images the induced latent space displacements by using shape and appearance mapper networks

- Deformation network similar to deformable NeRFs (instance specific network) in Edit-NeRF

Diffusion NeRF models

Diffusion NeRF models by training NeRF models from scratch on images generated from diffusion methods, which can generate views of the same object under different poses with appropriate prompting

- Dreamfusion: text to 3D diffusion NeRF model
    - Imagen diffusion model + mip- NeRF from scratch
- Latent NeRF: output 64x64x4 latent features that Stable diffusions could use (512x 512 x 3 RGB images)
    - Can do text and shape guidance for further shape refine and for strict shape constraint

- Magic3D (built on DreamFusion): targeted issue caused by low resolution diffusion images with two stage coarse- fine approach
    - Coarse: InstantNGP as NeRF trained on images from text prompts using image diffusion model eDiff-I
    - Geometry was then placed on a mesh and optimized in fine stage using images generated with latent diffusion model
    - Prompt based scene editing, personalized text to 3D generation via conditioning on image of a subject, style guided text to 3D generation
- RealFusion: single shot learning and used similar ideas as above
    - Used single image textual inversion as substitute for alternate views
    - Augmenting the input 2D image and associating it with a new vocabulary token to optimize the diffusion loss
- Diffusion models for single view scene learning via image conditioning: NeuralLift-360, NeRFdi, NerfDiff, DiffusioNeRF

Unbounded Scene and Scene Composition
Addressed issues with outdoor scenes

- Separate foreground and background
- Image by image variation in lighting and appearance
- Latent conditioning via image by image appearance codes to resolve this issue + semantic/ instance segmentation

- NeRF in the Wild
    - Real life photographs of same scene can contain per image appearance variations due to lighting and transient objects
    - Density MLP was kept fixed for all images
    - Conditioned color MLP on a per image appearance embedding
    - Another MLP conditioned on per image transient embedding predicted color and density functions of transient objects
    - Latent embeddings constructed using Generative Latent Optimization
- NeRF++: generate novel views for unbounded scenes by separating the scene using a sphere
    - Inside of sphere contained foreground object and fictitious camera views whereas background was outside the sphere
    - Outside of sphere re-parameterized using inverted sphere space
    - Two NeRFs, one inside and one outside + camera ray integral evaluated in two parts
    - How NeRF resolves shape- radiance ambiguity:
        - Wrong density configuration results in color configurations with high frequency components with respect to viewing angles
        - However, by construction (use of lower frequency components + view angles), NeRF models produce smoother color configurations
- GIRAFFE similar to NeRF-W using generative latent codes and separating background/ foreground:

    - Based on GRAF
    - Background treated as another object
    - 2D CNN discriminator with synthesized image resulting in a disentangled latent space for fine control over scene generation
- Fig-NeRF: took on scene composition but focused on object interpolation and amodal segmentation
    - Two separate NeRF models: one foreground (deformable Nerfies), one background (appearance latent code conditioned NeRF)
    - Two photometric losses, one for background, one for foreground
- Composition model to edit objects within scene
    - Voxel based approach which is jointly optimized with MLP parameters
    - Two NeRFs: one for objects, one for scene both conditioned on interpolated voxel features
    - Object NeRF further conditioned conditioned on a set of object activation latent codes
- NeRFReN: addressed problem of reflective surfaces in NeRF view synthesis
    - Separate radiance field into two components: transmittance and reflected
    - Depth smoothness loss, bidirectional depth consistency loss

Pose estimation

NeRF needs both input images and camera poses

- Originally unknown poses were acquired by COLMAP library
- Models which perform both pose estimation and implicit scene representation are formulated as offline structure from motion problems
    - Bundle adjustment is used to jointly optimize pose and the model
    - Or as an online simultaneous localization and mapping problem (SLAM)

- iNeRF: formulated pose reconstruction as inverse problem
    - Given pretrained NeRF, using the photometric loss, optimize the pose instead
    - Interest point detector and performed interest region based sampling
- NeRF-: jointly estimated NeRF model and camera parameters
- Bundle adjusted Neural Radiance Field (BARF): also jointly estimated poses alongside training NeRF
    - Coarse to fine registration by adaptively masking positional encoding (similar to nerfies)
- SCNeRF: self calibrating joint optimization model
    - Optimize not only unknown poses but also camera intrinsics for non linear camera models
    - Curriculum learning to gradually non linear camera/ noise parameters to joint optimization
- GNeRF: pose as generative latent code
    - GAN for NeRF images
    - Inversion network which uses generated image and output a pose which was compared to sample pose

- GARF: gaussian activation as an alternative to positional encoding along with bundle adjustment for pose estimation

NeRF and SLAM
- iMAP: first NeRF based dense online SLAM model
  - Jointly optimize camera pose and implicit scene representation using continual online learning
- NeRF- SLAM improved on existing by using Instant NGP as NeRF model
- NICE- SLAM: used hierarchical grid based representation of scene geometry

Adjacent methods for neural rendering

Explicit methods: Fast MLP-less volume rendering
- Plenoxel, HDR Plenoxel (based on Plenoctree): voxelized scene and store scalar for density and spherical harmonics coefficients direction dependent color
  - Skipped MLP and fit features on voxel grid
  - Primary contribution of NeRF is volumetric rendering of new view given densities and colors and not density and color MLPs themselves
- DVGO: directly optimize voxel grid of scalars for density
  - Use n- dimensional features and a decoding MLP
- TensoRF: scalar density and vector feature in 3D voxel grid, represented as a rank 3 and 4 tensor
  - Factorized tensors which decreased memory requirement of Plenoxels
- Streaming radiance fields: explicit representation method that specifically targeted NeRF training from video

Other Neural Volume Rendering
- IBRNet: for a target view, it selected N views from the training set whose viewing directions are most similar and processed those
  - Used a ray Tansformer
- Scene rendering transformer: CNN to extract feature patches from scene images which were then fed into an Encoder Decoder Transformer
  - Entire ray was queried at once
  - Geometry free and did not produce scene's density function or rely on geometric inductive biases
- NeRFormer: also uses Transformers as part of volume rendering

Applications
- Localization and navigation aspect and demonstrated a real life application of a pretrained NeRF
- Dex-NeRF: use NeRF's learned density to help robots grasp objects, specifically focusing on transparent objects
  - Evo-NeRF improves on this

- Urban NeRF models
    - Outdoor environments are unbounded; camera poses lack variety; large- scale scene are desired

- Urban Radiance Fields: in addition to standard photometric loss, also used LiDAR based depth loss and sight loss, skybox based segmentation loss
    - Skybox loss: if ray goes through a sky pixel where sky pixels are segmented through a pretrained model, forces point samples along rays through the sky pixels to have zero density
    - Depth loss: forces estimated depth to match LiDAR acquired depth
    - Sight loss: forces radiance to be concentrated at the surface
- Mega- NeRF: large scale urban reconstruction
    - NeRF++ inverse sphere parameterization to separate foreground and background but use ellipsoid instead
    - Incorporated per image appearance embedding code of NeRF-W
    - Also cached coarse rendering of densities and colors into an octree
        - Coarse initial view quickly produced but dynamically refined via more model sampling
- Block-NeRFs: city scale NeRF reconstruction from 2.8M street level images
    - Each individual block nerf was built on mip-NeRF by using its IPE and NeRF-W by using its appearance latent code optimization
    - Semantic segmentation to mask out transient objects
    - Visibility MLP to discard low visibility Block-NeRFs
    - Blocks were assigned with overlap and images sampled from overlapping Block-NeRFs
- S-NeRF and BungeeNerf from remote sensing images

- Human Faces and Avatars and Articulated Objects
    - Reconstruction of human avatars, finding applications in virtual/ augmented reality, digital entertainment, and communication
    - Reconstruction of human faces requires NeRF model to be robust under changes of facial expression which may manifest themselves as topological changes
        - Deformation field as additional MLP potentially conditioned by latent codes (deformation from baseline human face)
    - Reconstruction of human body poses requires NeRF to be robust under pose changes for articulated bodies (deformation field with template human body)

- NeRFies: NeRF model with a deformation field which improves performance in presence of non- rigid transformations in the scene (dynamic)
    - Mapped input observation frame coordinates to deformed canonical coordinates
    - Adding elastic regularization, background regularization, coarse to fine deformation regularization by adaptive masking the positional encoding
    - Also NerFace

- HyperNeRF (built on NeRFies): extend canonical space to higher dimension, adding additional slicing MLP which describes how to return to 3D representation using ambient space coordinates
- CoNeRF (built on HyperNeRF): controllable photo editing via sliders
- RigNeRF: deformation field guided by morphable 3D face model
- Neural Body:
    - Input video to anchor a vertex based deformable human body model (SMPL)
    - Onto each vertex, a latent code
    - Human pose parameters were then used to deform human body model
- A-NERF, HumanNeRF, etc.
- PREF: regularizing estimated motion conditioned on latent embedding
- LISA: modeled hands

Image Processing

- RawNeRF: adapting Mip-NeRF to high dynamic range image view synthesis and denoising
    - Renders in linear color space using raw linear images as training data: varying exposure and tone mapping curves
- HDR- NeRF: use low dynamic range training images with variable exposure time as opposed to raw linear images
- DeblurNeRF, NeRF SR, etc. focus on fundamental image processing tasks such as denoising, deblurring, super resolution

- Semantic NeRF: NeRF capable of synthesizing semantic labels for novel views
- Panoptic NeRF: 3D to 2D label propagation
- Panoptic Neural Fields: separate static objects from moving objects in scene
    - Background + foreground used two radiance fields
    - Moving objects each had their own radiance field
- Distilled knowledge of 2D feature extractor into 3D feature fields which they optimized in conjunction with in-scene radiance fields to produce NeRF model with semantic understanding
- SS-NeRF: encoding function, two position decoding functions (direction dependent + direction independent)
    - Network trained to produce a variety of scene properties such as color, semantic labels, shading, etc.
    - Scene property synthesis is achievable via volume rendering and simple NeRF training

Surface Reconstruction

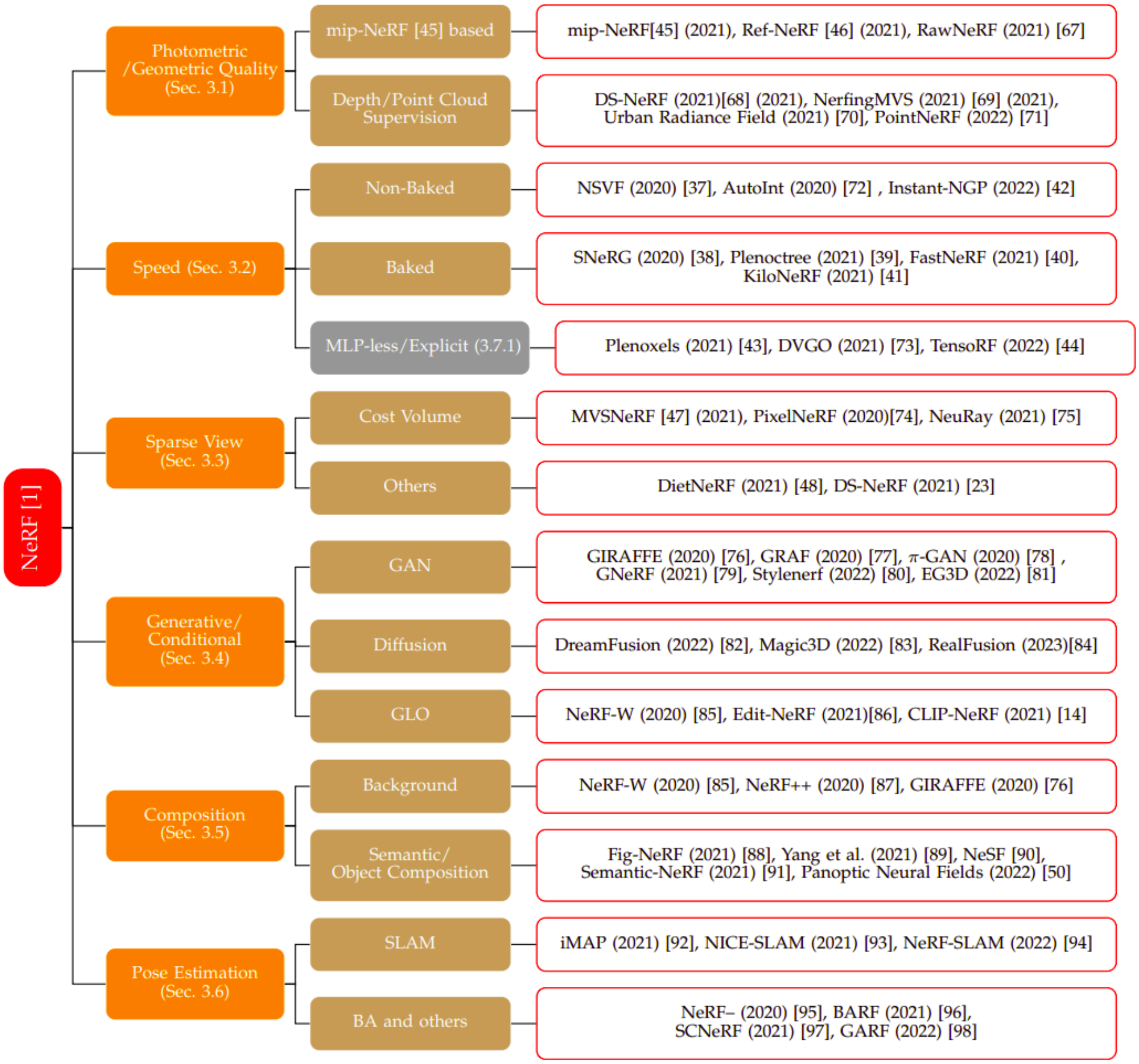
- UNISURF: reconstructed scene surfaces by replacing alpha value at ith sample point used in the discretized volume rendering equation with a discrete occupancy function
    - Surfaces retrieved via root finding along rays
- Neural Surface (NeuS): used signed distance field to define scene geometries instead of density

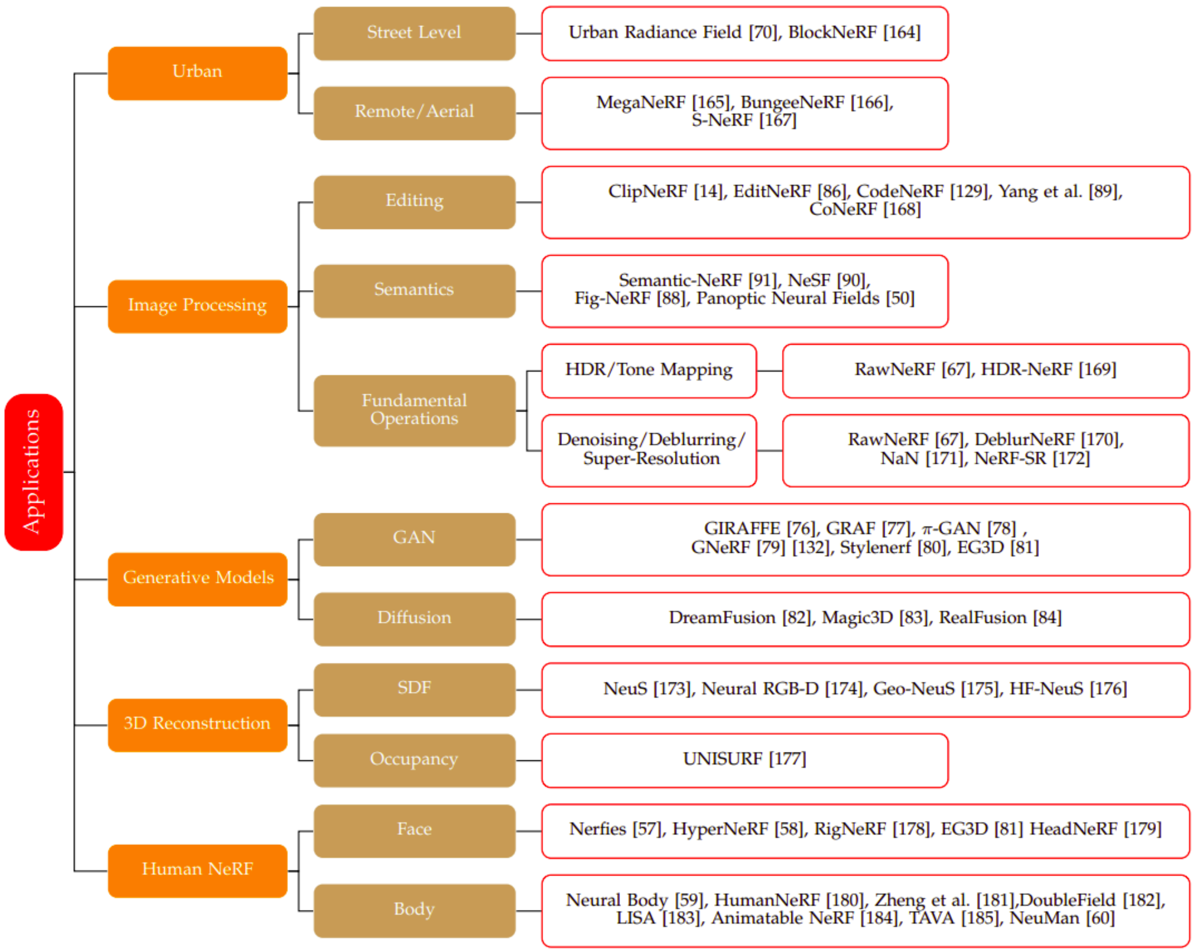
- HF- NeuS (improve on NeuS): separating low frequency details into a based SDF and high frequency details into a displacement function
- GeoNeuS: introduced multi view constraints in the form of multi view geometry constraint for the SDF supervised by sparse point cloud and multi view photometric consistency constraint
- SparseNeus: focusing on sparse view SDF reconstruction using geometry encoding volume with learnable image features
- Truncated SDF MLP by computing pixel color as weighted sum of sampled colors

Discussion

- Improving Speed
    - Baked models: Baked models cache the results of an already trained NeRF model, but do not improve training time.
    - Hybrid scene representation: Hybrid models separate learned scene feature from the color and density MLPs and use additional learned voxel/spatial-tree features. While this approach can speed up inference, it requires additional memory.
    - Explicit scene representation: Explicit models perform volume rendering directly on learned spatial features such as voxels without use of MLPs. This approach can also speed up inference, but requires additional memory. To further improve speed-based methods, researchers can use advanced volume rendering techniques such as empty space skipping and early ray termination. They can also lower the number of views and ray samples required for training and rendering.
    - TensoRF (reduce memory) and Instant NGP (speed up) show a lot of promise in this area
    - Researchers focus on improving data structure and design of additional learned scene features to account for memory trade off in hybrid and explicit scene methods
        - Improving sparse view capabilities
- Concerning quality
    - NeRF-W and GRAF with per image transient latent codes and appearance codes to control per image lighting and coloration changes and scene content
    - Mip- NeRF + 360: cone tracing
    - Ref- NeRF: further improved view dependence appearance, baseline
    - RawNeRF and DeblurNeRF for high quality denoising and deblurring
- Concerning pose estimation and sparse view
    - Non- SLAM pose estimation is a solved problem
    - NeRF based SLAM is a relatively under explored area of research
        - iMAP and Nice SLAM offer excellent NeRF based SLAM frameworks which could integrate faster and better quality NeRF models
        - Navigation inside NeRF environment is under explored

    - Sparse view and few shot seems to be solved but could be sped up for real time NeRF
        - Single shot NeRF by using image/ latent diffusion methods
- Concerning applications
    - NeRF has immediate applications in novel view synthesis and 3D reconstruction of urban environments and human avatars
    - Dividing environment into separate small scenes and representing each with small scene specific NeRF
    - Facilitating extraction of 3D mesh, point cloud or SDF from density MLPs and faster NeRF
    - Human avatar rendering and view synthesis + diffusion NeRF
    - Text to 3D NeRF models with scene and object editing
    - Few shot/ single shot NeRF
    - Semantic and object segmentation is active area of NeRF
        - 3D to 2D label transfer
    - Combining SLAM based NeRf with semantic understanding could be very good


NeRF [1]
Photometric /Geometric Quality (Sec. 3.1)
mip-NeRF [45] based
mip-NeRF[45] (2021), Ref-NeRF [46] (2021), RawNeRF (2021) [67]
Depth/Point Cloud Supervision
DS-NeRF (2021)[68] (2021), NerfingMVS (2021) [69] (2021), Urban Radiance Field (2021) [70], PointNeRF (2022) [71]
Speed (Sec. 3.2)
Non-Baked
NSVF (2020) [37], AutoInt (2020) [72] , Instant-NGP (2022) [42]
Baked
SNeRG (2020) [38], Plenoctree (2021) [39], FastNeRF (2021) [40], KiloNeRF (2021) [41]
MLP-less/Explicit (3.7.1)
Plenoxels (2021) [43], DVGO (2021) [73], TensoRF (2022) [44]
Sparse View (Sec. 3.3)
Cost Volume
MVSNeRF [47] (2021), PixelNeRF (2020)[74], NeuRay (2021) [75]
Others
DietNeRF (2021) [48], DS-NeRF (2021) [23]
Generative/ Conditional (Sec. 3.4)
GAN
GIRAFFE (2020) [76], GRAF (2020) [77], π-GAN (2020) [78] , GNeRF (2021) [79], Stylenerf (2022) [80], EG3D (2022) [81]
Diffusion
DreamFusion (2022) [82], Magic3D (2022) [83], RealFusion (2023)[84]
GLO
NeRF-W (2020) [85], Edit-NeRF (2021)[86], CLIP-NeRF (2021) [14]
Composition (Sec. 3.5)
Background
NeRF-W (2020) [85], NeRF++ (2020) [87], GIRAFFE (2020) [76]
Semantic/ Object Composition
Fig-NeRF (2021) [88], Yang et al. (2021) [89], NeSF [90], Semantic-NeRF (2021) [91], Panoptic Neural Fields (2022) [50]
Pose Estimation (Sec. 3.6)
SLAM
iMAP (2021) [92], NICE-SLAM (2021) [93], NeRF-SLAM (2022) [94]
BA and others
NeRF– (2020) [95], BARF (2021) [96], SCNeRF (2021) [97], GARF (2022) [98]


Applications
Urban
Street Level
Urban Radiance Field [70], BlockNeRF [164]
Remote/Aerial
MegaNeRF [165], BungeeNeRF [166], S-NeRF [167]
Image Processing
Editing
ClipNeRF [14], EditNeRF [86], CodeNeRF [129], Yang et al. [89], CoNeRF [168]
Semantics
Semantic-NeRF [91], NeSF [90], Fig-NeRF [88], Panoptic Neural Fields [50]
Fundamental Operations
HDR/Tone Mapping
RawNeRF [67], HDR-NeRF [169]
Denoising/Deblurring/ Super-Resolution
RawNeRF [67], DeblurNeRF [170], NaN [171], NeRF-SR [172]
Generative Models
GAN
GIRAFFE [76], GRAF [77], π-GAN [78] , GNeRF [79] [132], Stylenerf [80], EG3D [81]
Diffusion
DreamFusion [82], Magic3D [83], RealFusion [84]
3D Reconstruction
SDF
NeuS [173], Neural RGB-D [174], Geo-NeuS [175], HF-NeuS [176]
Occupancy
UNISURF [177]
Human NeRF
Face
Nerfies [57], HyperNeRF [58], RigNeRF [178], EG3D [81] HeadNeRF [179]
Body
Neural Body [59], HumanNeRF [180], Zheng et al. [181],DoubleField [182], LISA [183], Animatable NeRF [184], TAVA [185], NeuMan [60]