Journal Article Summaries

Jeffrey Liang

May 10, 2025

Summary 1: MeshSDF: Differentiable Iso-Surface Extraction

Citation: Remelli *et al.*, 2020
Summary: Using a neural network to model a signed distance function, we sample points from the zero-level set of the watertight shape and then use Marching Cubes (MC) to create a mesh between points. Rather than differentiating with respect to the MC mesh, we differentiate directly with the neural network since the loss function only cares about zero-level set. There do exist differentiable MC variants but they can be complex/apply only in certain conditions.
Key Points:

- Vertices are sampled from zero level set usually, so if the loss function only cares about the zero level set then you can directly differentiate with respect to the neural net. When the loss function cares about the facets, then we need a differentiable Marching Cubes (but the points are still sampled from the zero level set).
- When a point s undergoes an infinitesimal perturbation Δs, the local surface is perturbed in the opposite direction of the surface normal: $\frac{\partial v}{\partial s} v = -n(v) = -\nabla s(v)$.
- a negative Δs causes points with slightly positive distance values move closer to the zero set (as you subtract), that is, the surface is inflated.
- Marching Cubes (MC) converts implicit functions to 3D surface meshes $\mathcal{M} = (V, F)$. It samples the network on a discrete 3D grid, detecting zero-crossing of the field along grid edges and builds a surface mesh with a lookup table. The position of vertices on grid edges involves linear interpolation, which is modelled by a function discontinuous at $s_i = s_j$, so cannot allow topology changes through backpropagation.
- This method is used for tasks where we deform a 3D mesh $\mathcal{M} = (V, F)$, where $V = \{\boldsymbol{v}_1, \boldsymbol{v}_2, ...\}$ denotes vertices in $\mathbb{R}^3$ and F facets to minimise a task-specific loss function $\mathcal{L}_{\text{task}}(\mathcal{M})$.
- Signed distance function (SDF): $s : \mathbb{R}^3 \to \mathbb{R}$. Neural network f_θ is trained on watertight surfaces $\mathcal{S}$ to approximate SDF s by minimising

$$\mathcal{L}_{\text{sdf}}\left(\{\boldsymbol{z}_S\}_{S\in\mathcal{S}}, \theta\right) = \sum_{S\in\mathcal{S}} \frac{1}{|X_S|} \sum_{\boldsymbol{x}\in X_S} |f_\theta(\boldsymbol{x}, \boldsymbol{z}_S) - s(\boldsymbol{x})| + \lambda_{\text{reg}} \sum_{S\in\mathcal{S}} \|\boldsymbol{z}_S\|_2^2$$

  where $\boldsymbol{z}_S \in \mathbb{R}^Z$ is a Z-dimensional encoding of surface S, θ denotes network parameters, X_S represents 3D point samples we use to train our netowrk, and λ_{reg} affects regularisation.

  When given a mesh, we want to be able to evaluate:

$$\frac{\partial \mathcal{L}_{\text{task}}}{\partial \boldsymbol{z}} = \sum_{\boldsymbol{v}\in V} \frac{\partial \mathcal{L}_{\text{task}}}{\partial \boldsymbol{v}} \frac{\partial \boldsymbol{v}}{\partial f_\theta} \frac{\partial f_\theta}{\partial \boldsymbol{z}}.$$

  As f_θ approximates an SDF, we can replace $\frac{\partial \boldsymbol{v}}{\partial f_\theta}$ with $-\nabla f_\theta(\boldsymbol{v}, \boldsymbol{z})$.

- Two loss functions are considered between surfaces S and T:

$$\mathcal{L}_{\text{task1}} = \min_{s \in S} d(s, T) + \min_{t \in T} d(S, t), \qquad \text{surface-to-surface distance}$$
$$\mathcal{L}_{\text{task2}} = \|\text{DR}(S) - \text{DR}(T)\|_1 \qquad \text{image-to-image distance}$$

- For shape optimisation, we can use the following loss function (aerodynamics):

$$\mathcal{L}_{\text{task}}(\mathcal{M}) = \iint_{\mathcal{M}} g_\beta \boldsymbol{n}_x \, \mathrm{d}\mathcal{M} + \mathcal{L}_{\text{constraint}}(\mathcal{M}).$$

Summary 2: Structure-from-Motion Revisited

Citation: Schönberger and Frahm, 2016
Summary: A new (incremental) SfM technique which looks to address image registration and triangulation more efficiently and robustly. It augments scene graph using geometric verification strategy. Next best view selection for incremental reconstruction process. Robust triangulation to produce more complete scene structure with reduced computational cost. An iterative BA, re-triangulation, and outlier filtering strategy to improve completeness. More efficient BA parametrisation for dense photo collections.
Key Points:

- SfM starts with correspondence search given a set of images $\mathcal{I} = \{I_i \mid i = 1, ..., N_I\}$:
  1. Feature extraction: $\mathcal{F}_i = \{(x_j, \boldsymbol{f}_j) \mid j = 1, ..., N_{F_i}\}$ at location $\boldsymbol{x}_j \in \mathbb{R}^2$
  2. Matching ($\mathcal{O}(N_I^2 N_{F_i}^2)$): find most similar feature in the other image through brute force to give set of potentially matching image pairs $\mathcal{C} = \{\{I_a, I_b\} \mid I_a, I_b \in \mathcal{I}, a < b\}$ and associated feature correspondences $\mathcal{M}_{ab} \in \mathcal{F}_a \times \mathcal{F}_b$.
  3. Geometric verification: estimate projection transformation to map between feature points (use homography or epipolar geometry) + RANSAC. Outputs a scene-graph, with images as nodes and verified pairs of images as edges.
- Incremental reconstruction: outputs pose estimates $\mathcal{P} = \{\boldsymbol{P}_c \in \boldsymbol{SE}(3) \mid c = 1, ..., N_P\}$ for registered images and the reconstructed scene structure as a set of points $\mathcal{X} = \{\boldsymbol{X}_k \in \mathbb{R}^3 \mid k = 1, ..., N_X\}$.
  1. Initialisation: done with carefully selected two-view reconstruction - can be from densse location in the image graph with many overlapping cameras.
  2. Image registration: can register to current model by solving Perspective-n-Point (PnP) problem suing using correspondences to triangulated points already in the image. Each new registered image extends the set $\mathcal{P}$. This paper attempts to improve in this part.
  3. Triangulation: a new scene point can be triangulated and added to $\mathcal{X}$ as soon as one more image also covering the same scene but from a different viewpoint is registered. The redundancy and additional correspondences increases stability. This paper addresses more robust/efficient method.
  4. Bundle adjustment: A joint non-linear refinement of camera parameters $\boldsymbol{P}_c$ and point parameters $\boldsymbol{X}_k$ which minimise the reprojection error $E = \sum_j \rho_j \left(\|\pi(\boldsymbol{P}_c, \boldsymbol{X}_k) - \boldsymbol{x}_j\|_2^2 \right)$, where π projects scene points to the image space and ρ_j is a loss function to down-weight outliers. Levenberg-Marquardt is used to solve BA problems. Can use Schur complement trick. This paper attempts to improve on this part.

- Current challenges for SfM: failure to register images which should be, produce broken models due to mis-registrations or drift. Caused by incomplete scene graphs, or in failures in reconstruction stage due to inaccurate scene structure.

Summary 3: CAPNet: Continuous Approximation Projection for 3D Point Cloud Reconstruction Using 2D Supervision

Citation: Navaneet *et al.*, 2019
Summary: Constructs a 3D point cloud by projecting them into multiple 2D views and comparing with ground-truth 2D masks in each view (weak supervision). A continuous approximation of points in the point cloud produces smooth projections in a differentiable manner. Uses encoder-decoder architecture.
Key Points:

- Point cloud representations are better than volumetric representations since the latter suffers from information sparsity. Surface voxels provide structural information, however internal voxels increase the computational complexity with minimal addition to information.
- *Goal*: From a single image of an object, reconstruct a 3D point cloud representation of the object. If I is an image from the training set and f is a trained network, let $p = f(I)$ be the corresponding 3D point cloud reconstruction. A projection $P(p, v)$ is from an arbitrary view point v is obtained by performing a perspective transformation and projecting the transformed point cloud onto a plane. The transformed point $\hat{p}_n = (\hat{x}_n, \hat{y}_n, \hat{z}_n)$ in camera coordinates is obtained from: $\hat{p}_n = K(R_v p_n + t_v) \quad \forall n \in \{1, ..., N\}$, where K, R_v, t_n are the camera intrinsics and extrinsics. Training uses ground truth 2D masks M to supervise projection $\hat{M} = P(p, v)$.
- The encoder takes in a 2D image and the decoder reconstructs the point cloud. To compute the loss, the predicted point cloud is projected from V different view-points and compared with corresponding ground truth projections.
- The final loss function is $\mathcal{L} = \mathcal{L}_{bce} + \lambda \cdot \mathcal{L}_{\text{aff}}$. Let $\hat{M}^v_{i,j} = \tanh\Big(\sum_{n=1}^{N} \phi(\hat{x}_n - i) \cdot \phi(\hat{y}_n - j)\Big)$, where ϕ is the kernel function $\phi(k) = \exp\Big(\frac{-k^2}{2\sigma^2}\Big)$ (this Gaussian kernel allows smooth, accurate projections with no holes). The loss function is binary cross-entropy loss:

$$\mathcal{L}_{bce} = \sum_{v=1}^{V} -M^v \log\Big(\hat{M}^v\Big) - (1 - M^v) \log\Big(1 - \hat{M}^v\Big)$$

  where M^v and $\hat{M}^v$ are the ground truth and predicted masks, respectively, of dimension (H, W). This loss function results in reconstructions with many outlier points, so a *nearest point affinity loss* is imposed which minimises the nearest neighbour distance between two pixel maps weighted by pixel confidence:

$$\mathcal{L}_{\text{aff}} = \sum_{v=1}^{V} \sum_{i,j}^{H,W} \min_{(k,l) \in M^v_+} ((i-k)^2 + (j-l)^2) \hat{M}^v_{i,j} M^v_{k,l} + \sum_{v=1}^{V} \sum_{i,j}^{H,W} \min_{(k,l) \in \hat{M}^v_+} ((i-k)^2 + (j-l)^2) M^v_{i,j} \hat{M}^v_{k,l}$$

  where M^v_+ and $\hat{M}^v_+$ are sets of pixel coordinates of the ground truth and predicted projections whose values are non-zero.
- The resultant point cloud is evaluated with the ground truth by computing the Chamfer distance:

$$d_{\text{Chamfer}}(\hat{P}, P) = \sum_{x \in \hat{P}} = \min_{y \in P} \|x - y\|_2^2 + \sum_{y \in P} = \min_{x \in \hat{P}} \|y - x\|_2^2.$$

Ground truths were obtained by randomly sampling 16,384 points on the object surface and then performing farthest point sampling to obtain 1024 points.

Summary 4: Occupancy Networks: Learning 3D Reconstruction in Function Space

Citation: Mescheder *et al.*
Summary: Uses a neural network classifier to represent the continuous decision boundary of a 3D surface, enables occupancy to be evaulated at arbitrary resolution.
Key Points:

- Existing 3D representations can be broadly categorised into three categories: voxel-based representations, point-based representations, and mesh representations. Point clouds lack connectivity structure of underlying mesh and hence require extra post-processing steps to extract 3D geometry.
- Let the occupancy function $o : \mathbb{R}^3 \to \{0, 1\}$ denote whether that point is occupied by the object. The paper's occupancy function outputs a probability between 0 and 1 for every point $p \in \mathbb{R}^3$. We want to condition the reconstructed 3D object output on the input $x \in \mathcal{X}$. Hence, we train an *occupancy network* $f_\theta : \mathbb{R}^3 \times \mathcal{X} \to [0, 1]$, which takes a pair (p, x) and outputs a real number representing the probability of occupancy.
- We randomly sample points in the 3D bounding volume of the object. The loss function for a batch $\mathcal{B}$ is:

$$\mathcal{L}_{\mathcal{B}}^{\text{gen}} = \frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} \left[\sum_{j=1}^{K} \mathcal{L}(f_\theta(p_{ij}, z_i), o_{ij}) + \text{KL}(q_\psi(z|(p_{ij}, o_{ij})_{j=1:K}) \| \, p_0(z)) \right]$$

- Creates a mesh using coarse voxel grid and evaluating occupancy network for cells in this grid. For boundary voxels, we further subdivide them and repeat, until we reach the desired resolution, before applying Marching Cubes.
- In ShapeNet results, able to construct finer details which get lost in coarser voxel representations.

Summary 5: DRWR: A Differentiable Renderer without Rendering for Unsupervised 3D Strucutre Learning

Citation: Han *et al.*, 2020
Summary: An unsupervised differentiable renderer uses losses to evaluate the alignment of projections of reconstructed 3D point clouds with ground truth object silhouettes, instead of pixel-wise losses. Idea is to pull points to the surface of the silhouette and then push them far away from each other.
Key Points:

- This is unsupervised, so only requires images for training, whereas existing approaches require large amounts of data.
- Existing pixel-wise loss functions render images from 3D structures but require rasterisation or pixel-wise interpolation commonly, which they claim don't actually contribute to structure learning
- DRWR: doesn't require visibility handling, shading, or pixel-wise interpolation.

- Loss function: smooth unary loss pulls projection of each 3D point into the foreground (image we're interested in) and binary/pairwise loss pushes each pair of projections lying inside the foreground far away (structure-aware since only considers repulsion once both points appear in foreground). This ensures the entire foreground is covered and prevents clumping.
- Link to my approach: renderers bridge the gap between 3D and 2D by using 2D loss functions on 3D structure, loss functions which are usually defined on RGB pixelvalues or pixel-wise silhouette coverage. Renders predicted 3D structure from specific view angle into an image.

Summary 6: Deep Declarative Networks: A New Hope

Citation: Gould, Hartley, and Campbell, 2020
Summary: Networks are defined in terms of desired behaviour rather than an explicit forward function, with the desired behaviour implicitly defined as a solution to an optimisation problem.
Key Points:

- We only need the optimal solution to an optimisation problem and twice-differentiable objective and constraint functions to perform backpropagation - the method used to find the solution can be non-differentiable such as RANSAC.
- Identities and solutions use the Schur complement and block matrix inverses (on Wikipedia).

Summary 7: Small Steps and Level Sets: Fitting Neural Surface Models with Point Guidance

Citation: Koneputugodage *et al.*, 2024
Summary: Spherical initialisations of point clouds for neural SDFs often get trapped in local minima due to the biased initialisation. This approach uses intermediary "guiding points" during optimisation to allow the point cloud to fit complex shapes. Uses percentage of self-intersecting outward normals (SION%) to provide a metric of the diffence between surface geometries of the current and target shapes.
Key Points:

- The idea behind SION% is that if there are large concavities in the target shape, the surface normals will self-intersect and few will make it out to the current surface which doesn't fit inside the cavity - however, if it did "fit" inside, then the surface normals would not self-intersect.
- Performance decreases on IoU and Chamfer distance metrics as the shapes get more complex (increasing SION%), though at much slower rate than other SotA.
- In each iteration, the guiding points are determined by taking the guiding points from the previous iteration by moving it towards the closest target point.

Summary 8: DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation

Citation: Park *et al.*, 2019
Summary: Revolutionary paper about using neural networks to implicitly model the signed distance

function of a 3D watertight shape.
Key Points:

- Trained using set of pairs X composed of 3D point samples and the SDF value, $X := \{(\boldsymbol{x}, s) \mid SDF(\boldsymbol{x}) = s\}$, such that the trained network is a good approximator of the SDF, that is, $f_\theta(\boldsymbol{x}) \approx SDF(\boldsymbol{x})$. Loss is an L_1 loss function which uses clamping.
- Auto-decoder which has the latent code as input and directly optimises that, involves taking the maximum-a-priori estimate of the loss function + a regularising term over all points over all shapes
- To make the network flexible and adaptable to many shapes, a latent code describing the shape is inputted with the 3D point into the network, rather than a single point input with the network containing information about the shape.
- The SDF value is the signed distance from the shape's surface, with negative values indicating the point is inside the shape, zero values meaning the point lies on the surface, and positive values meaning the point is outside the shape. The magnitude of the value is the distance to the surface.

Summary 9: Multi-view Convolutional Neural Networks for 3D Shape Recognition

Citation: Su et al., 2015
Summary: Classifies 3D objects using 2D projections of said object from different views using a CNN, outperforms previous approaches which had access to the 3D information.
Key Points:

- They can outperform 3D based methods as 1) 2D representations are less memory expensive compared to voxel representations and thus training is more efficient, 2) image descriptors have advanced to be more information dense, 3) there exist massive image databases which are well-annotated
- Rather than averaging views, the multi-view CNN is able to identify more informative views and combine these to form the 3D object (think about the pooling layers).
- "Jittering" used to augment data and provide analogous samples to changing the view.

Summary 10: GRF: Learning a General Radiance Field for 3D Representation and Rendering

Citation: Trevithick and Yang, 2021
Summary: Takes a set of 2D images with known camera poses and intrinsics, constructs 3D rendering from arbitrary positions using neural network, uses attention mechanism from differing views to account for occlusions.
Key Points:

- Builds upon NeRF, can infer from unobserved viewpoints. Can generalise unlike NeRF to any set of images to a 3D structure with geometric details.
- Exploits the underlying 3D geometry unlike previous approaches which generated images from unseen views by learning a 2D manifold.
- Four components: 1) feature extractor from 2D pixels using CNN, 2) reprojector to transform

into 3D space, 3) aggreggator to obtain general features for 3D points, 4) NeRF to render internal representation

Summary 11: Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images

Citation: Wang et al., 2018
Summary: Graph-based CNN which learns deformations (not the structure itself) of an ellipsoid tempplate to output a 3D mesh from a single colour image.
Key Points:

- Coarse-to-fine strategy for stability by graph unpooling (typical division strategy used in computer graphics where new vertex is midpoint of an edge, generating 4 new triangles), uses many surface and geometric losses to improve realism and fidelity
- graph-based convolutional network used to model the mesh with nodes = vertices and connections = edges, use a pooling layer of 2D images in the GCN to incorporate features/knowledge in 3D construction
- Multi-view geometry is restricted by the coverage of views (cannot generalise to unseen viewpoints) and appearance (cannot generate non-Lambertian surfaces), hence why learning approaches are important

Summary 12: AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation

Citation: Groueix et al., 2018
Summary: From a latent shape vector, sample a point on a unit square, the network produces a 3D point with the square deformed in 3D to be a continuous planar surface (locally 2D manifold) allowing arbitrary resolution, and repeat this for multiple patches.

Summary 13: Mesh R-CNN

Citation: Gkioxari, Malik, and Johnson, 2019
Summary: Mix of 2D perception and 3D shape prediction. Takes input image and predicts meshes of variable topology by first predicting coarse voxel structure, converting to mesh, and then refining the mesh with graph convolution network.

Summary 14: Deep Marching Tetrahedra: a Hybrid Representation for High-Resolution 3D Shape Synthesis

Citation: Shen et al., 2021
Summary: Generative model to generate 3D shapes via user guided input (such as coarse voxels), hybrid implementation where explicit representation used to allow finer geometric features to be rendered and implicit representation (deformable tetrahedral grid) to allow arbitrary topology.
Key Points:

- Uses Marching Tetrahedra algorithm which is differentiable unlike MC, adapts resolution by deforming the tetrahedra grid — the singularity for linearly interpolating the crossing remains but in practice only evaluated when vertex sign is different
- Input is point cloud or low-res voxel which gets converted into a feature vector of the 3D volume,

Summary 15: Shape as Points: a Differentiable Poisson Solver

Citation: Peng et al., 2021
Summary: Differentiable Poisson surface reconstruction method used in both optimisation and learning-based scenarios (noisy point clouds). Render using MC.
Key Points:

- Optimisation-based: noisy oriented point cloud as input, pass this into DPSR to obtain dense indicator grid, then sample points and use MC to generate a mesh which can be compared using Chamfer to target. MC not differentiable, but gradient approximated using inverse normals of mesh sampled vertices (from MeshSDF).
- Learning -based: learns features/normals from a point cloud using an MLP and then calculating an indicator grid using DPSR and comparing it to the target indicator grid to supervise learning.
- Doees not require normals like traditional PSR.
- Poisson equation intuition: the gradient of the indicator function image is approximated by point normals.
- Discretizes allowing inversion of divergence operator, use spectral methods to solve the equation (FFT highly optimised on GPUs).

Summary 16: Convolutional Occupancy Networks

Citation: Peng et al., 2020
Summary: A new implicit representation (compared to MLPs which fail to encode local information/structured reasoning) which uses convolutional encoders and then a decoder to get a probabilistic occupancy grid.
Key Points:

- Converts 3D input (point cloud or voxel) into features, run features through CNN onto occupancy grid output

Summary 17: Multiview Neural Surface Reconstruction by Disentangling Geometry and Appearance

Citation: Yariv et al., 2020
Summary: Separates geometry from appearance from camera parameters to construct 3D scenes from input images (even with variable lighting conditions).
Key Points:

- Input: masked 2D images (indicator function approximated using sigmoid for differentiability)

and camera estimates.

- Differentiable renderer which is able to render the surface of a shape represented by the zero level set of the implicit representation (neural network).
- Able to render material/colour alongside reconstruct shape.
- Uses differentiable ray casting, parametrises the intersection of the ray with the surface of the object with the geometry and camera parameters.

Summary 18: Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations

Citation: Sitzmann, Zollhofer, Wetzstein
Summary: Differentiable ray-marching algorithm trained on 2D images and camera poses, outputs 3D scene (geometry + appearance). Maps 3D world coordinate to feature representation of scene properties at that coordinate. Does not require 3D supervision.
Key Points:

- Can generate images without 2D convolutions - not a 3D shape reconstruction, but still involves a 3D implicit representation to generate images from novel views.

Summary 19: Differentiable Volumetric Rendering: Learning Implicit 3D Representations without 3D Supervision

Citation: Niemeyer et al., 2020
Summary: Differentiable renderer for implicit shapes (and texture). Depth gradients obtained with implicit differentiation. Input 2D posed images. Outputs depth map from occupancy network, which is unprojected into 3D and rendered to be compared with ground truth image.
Key Points:

- 2D supervision (depth maps or multi-vew images) attempts to overcome learning-based 3D reconstruction approaches which are restricted to synthetic data (and restricts their prior knowledge) during supervision. However many suffer from discretisation artifacts (meshes and voxels).
- Mesh based methods require deformable template mesh or a collection of 3D patches which leads to self-intersections and non-watertight meshes
- uses occupancy network to assign a probability of occupancy to each point, and texture (colour) uses a texture field
- able to learn without 3D ground truth data
- key contribution is analytical gradient to optimise for the surface depth and overcome non-differentiable argmin and iterative secant method

Summary 20: Octree Generating Networks: Efficient Convolutional Architectures for High-resolution 3D Outputs

Citation: Tatarchenko, Dosovitskiy, and Brox, 2017
Summary: Convolutional decoder which outputs higher resolution octree from high-level feature descriptions and reconstructing 3D objects from single images.
Key Points:

- Uses hash table rather than storing cell pointers using an index-value pair (m, v) where m generated from spatial coordinate and level using Z-order curves and v generated any signal.

Summary 21: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation

Citation: Qi et al., 2017
Summary: Takes point clouds and outputs classification, segmentation, and semantic parsing.
Key Points:

- Uses symmetric function of max pooling for permutation invariance (as point clouds are a set)
- Network learns set of functions which select informative points of the point cloud and encode reason for selection
- Network should be invariant to ordering and transformations such as rotation or translation

Summary 22: DUSt3R: Geometric 3D Vision Made Easy

Citation: Wang et al., 2024
Summary: Multi-view stereo reconstruction in the wild, from unconstrained image collection, can output pointmaps from which camera parameters, depthmaps, and 3D reconstruction can be recovered
Key Points:

- Unlike most approaches which first attempt to estimate camera parameters, this work performs regression of pointmaps to relax hard constraints of projective camera models
- Unconstrained image-based dense 3D reconstruction form multiple views subsumes many other geometric 3D vision tasks
- Attempts to overcome challenge of existing approaches which are only as good as the quality of input images and camera parameters
- Jointly processes input images and 3D pointmaps (dense 2D field of 3D points) in order to associate 2D structures with 3D shapes
- Need to apply MC, PSR, or other learning-based methods to extract mesh from pointmap
- From a pair of images, encodes them using ViT (patchify) with shared weights, but then decoded separately, with the second decoder also optimising for relative pose to image 1. Outputs point map and confidence map for each head.

Summary 23: NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis

Citation: Mildenhall et al., 2020
Summary: Trains one MLP per scene with input being single 5D coordinate (x,y,z and viewing direction θ, ϕ) and output is volume density and view-dependent radiance (colour and lighting). Loss between rendered images and observed image inputs.
Key Points:

- Uses positional encoding with sine and cosines to take input vectors into a higher frequency dimension, allowing finer details to be generated
- Estimates radiance using quadrature (whatever that means)

Summary 24: Pix2Vox++: Multi-scale Context-aware 3D Object Reconstruction from Single and Multiple Images

Citation: Xie et al., 2020
Summary: Takes multi-view images to produce coarse voxel strucures which are then fused adn refined

Summary 25: 3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction

Citation: Choy et al., 2016
Summary: Uses Recurrent NN to map image/s to reconstructed 3D shape in the form of a voxelised occupancy grid.
Key Points:

- Convolution NN with many LSTMs with restricted connections for encoding, decoder is 3D deconvolution network, uses softmax voxel-wise loss
-

Summary 26: BundleSDF: Neural 6-DoF Tracking and 3D Reconstruction of Unknown Objects

Citation: Wen et al., 2023
Summary: Near real-time tracking method of objects (6 DoF) from a video sequence whilst performing 3D reconstruction. Only need to be segmented in the first frame.
Key Points:

- 1) Match features between consecutive segmented video frames to obtain rough pose estimates of the object, some which are stored in memory. 2) Pose graph constructed from memory pool 3) Online optimisation of pose graph with current pose, then stored back in memory pool, 4) All posed frames in memory pool used to train Neural Object Field (geometry is an SDF and also visual appearance/texture)
- Uses octree for (hierarchical) ray sampling

Summary 27: TopoNet: Topology Learning for 3D Reconstruction of Objects of Arbitrary Genus

Citation: Charrada et al., 2022
Summary: Takes RGB image and reconstructs 3D mesh by deforming a template mesh and then performs face-pruning to refine the mesh and allow arbitrary genus changing using an RL network.
Key Points:

- Conv Graph Net to extract features of initial mesh and deform it
- Does not look watertight bro