Training-Time Uncertainty Estimation for Next Best View in Implicit 3D Reconstruction

Anonymous submission

Abstract

The emerging implicit 3D reconstruction recovers fine details of the 3D scene without dense data grids, but incurs difficulty for next best view (NBV) methods to automatically find a shortest sequence of views that optimize the process due to the absence of an explicit representation. Prior work attempt to resolve this difficulty by training an extra model head to predict the uncertainty of the model output as an approximation the potential information gain for NBV selection. However, we show that the uncertainty of the model output often mismatches with actual information gain in the cutting-edge grided implicit 3D reconstruction due to a new type of uncertainty caused from capacity limit of each grid unit and causes inefficiency. Instead of predicting uncertainty of the model output, in this paper, we propose an algorithm (TUEV) for NBV in implicit 3D reconstruction that tracks certainty of the grided latent codes purely based on aggregating the synaptic dynamics at training time. In this way, the potential information gain for grided implicit 3D reconstruction is better approximated with little computation burden (no inference needed as in prior work). Evaluation shows XX%X̃X% reduction of the length of views to reach the same reconstruction accuracy compared with the baselines. The source code for result reproduction is available at https://github.com/XXX/XXX.

Introduction

Next-Best-View (NBV) aims to find a shortest and most informative sequence of views (sensor positions and the captured images) in an unknown scene to boost efficiency and accuracy of 3D reconstruction, a robotic task fundamental for applications such as augmented/virtual reality and autonomous driving. This sequence is typically retrieved by iteratively estimating information gain of candidate views and adding the highest one to the sequence. While traditional NBV methods process an explicit 3D representation (e.g., point cloud) heuristically or with AI models to find most informative views (e.g., frontiers), the emerging implicit 3D reconstruction methods exploit implicit representation (implicit 3D reconstruction) to achieve finer textural information, scene completion, etc., but also incur difficulty for NBV due to the absence of explicit representation.

For implicit representation, a straight-forward estimation of information gain when inputting a candidate view for training is the reduction of uncertainty of model parameters, which is, however, reported to be computation expensive and intrusive to the model structure and training process. To tackle these problems, prior work propose to replace the estimation of information gain with the uncertainty of model outputs, which models how much the reconstructed 3D model disagree with the existing training data and implies opportunity for further improvement. They either online or offline train an extra model head to predict the uncertainty (e.g., standard deviation) of reconstructed 3D scene from a view and the most uncertain view is selected as NBV.

However, the cutting-edge grided implicit 3D reconstruction brings challenge to these methods, where they suffer frequent mismatch between estimated uncertainty of model output and actual information gain. Specifically, grided implicit 3D reconstruction decomposes the reconstructing 3D scene to a grid of small geometries and also decomposes the training model to a grid of latent codes, each mapping one small geometry, so that each latent code fits to the reconstructing geometry almost in real time during training and fine textural information is also extracted.

While these methods are good at discovering details in candidate views not attended by the training model, the grided implicit 3D reconstruction minimizes such uncertainty with decomposition, and further incurs a new type of uncertainty that they did not consider, uncertainty caused by capacity limit of local latent codes. These methods typically assume a simple reconstructing 3D scene and a model large enough to model the whole scene, which do not hold in grided implicit 3D reconstruction, where local latent codes could reach their capacity limit, causing high uncertainty of model output from these latent codes but limited information gain. As a result, it is frequently observed that the NBV selections of these methods were stuck at certain spots in our evaluations, causing inefficiency. Also, these methods require inference of the model to predict uncertainty for each candidate view, which is heavy when the number of candidate views increases for more accurate estimation, especially for edge devices exploring large scenes.

In this paper, we propose Training-time Uncertainty Estimation for next best View in implicit 3D reconstruction (TUEV). Assuming that local latent codes have limited capacity, instead of predicting output uncertainty, we propose to track the certainty of local latent codes for grided implicit 3D reconstruction. Although accurate estimation of uncertainty of model parameters is challenging and computation expensive as stated above, we manage to approximate the comparative relationship of the parameter uncertainty between the latent codes by aggregating and comparing their synaptic dynamics during training, which suffices for NBV selection. With the 3D correspondence of each grided latent code in the camera frustums of candidate views and the comparative relationship of their uncertainty, we are able to render the comparative relationship between candidate views without incurring any extra training of inference computation burden. As a result, TUEV is suitable for edge devices with limited computation resources such as robots where NBV for 3D reconstruction is typically deployed.

The main contributions of this paper are as follows:

- We propose a new uncertainty estimation method (TUEV) for NBV in implicit 3D reconstruction. Instead of the uncertainty of rendered output, we estimate the certainty of grided latent codes to get more accurate estimation of possible information gain of candidate views, boosting the efficiency and accuracy of implicit 3D reconstruction.
- We introduce an efficient algorithm that approximates the comparative relationship between the uncertainty of latent codes for NBV. Compared with prior work, our proposed algorithm directly operates on latent codes and incurs little computation burden, making it suitable for edge devices where NBV is typically deployed.
- Extensive experiments on various complex 3D models and large scale indoor scenes shows that TUEV reduces XX% XX% input views to reach the same reconstruction accuracy as the prediction-based methods. The source code to reproduce our results is publicly released at https://github.com/xxxxxx.

Background

Neural Implicit Representations

Neural implicit representation is a emerging mapping representation demonstrating promising results for object geometry reconstruction, scene completion, novel view synthesis and also generative modelling. They typically feature a Neural Radiance Field (NeRF) structure that learns a density and radiance field supervised by 2D views (camera position & orientation and the images captured) with an MLP model. iMAP uses a single MLP neural model as the underlying 3D scene representation and with a comparatively simple implicit representation and efficient rendering pipeline, iMAP achieves near real-time performance in training.

However, recent researches report a single MLP representation is not scalable due to limited capacity and tends to ignore high-frequency (complex) details. They propose to decompose the whole 3D scene to grided local scenes and train local implicit representations to learn the local geometry in each local scene, which improves the level of detail in reconstruction as each implicit representation only needs to map a local region rather than the geometry of a whole scene. Organizing the local implicit representations as a grid, we can easily find the 3D correspondence between the local implicit representations and the 3D scene, which is the basis of our proposed method.

Implicit 3D reconstruction

As recent researches decompose the implicit representation to grided local implicit representations, some also decompose the training pipeline to effectively leverage prior knowledge of local geometry embedded in MLP. Specifically, instead of training the whole MLP, they (grided implicit 3D reconstruction) separate the MLP model to an AutoEncoder-like network that consist of an encoder, grided latent codes and a decoder, with the encoder and the decoder pretrained over various scenes to extract generalizable knowledge of 3D reconstruction. Because they only need to optimize the local latent codes, they manage to reconstruct complex geometry of 3D scenes in a view with several training iterations, retaining real-time performance. We select one of these work, BNV-Fusion as our major research target, which takes depth images and camera poses as input and achieves high quality shape reconstruction of complex 3D scene.

Next Best View

Traditional NBV typically aims to find a shortest sequence of views from a set of candidate views that optimize the coverage of a previously unknown area. Given the existing partial explicit map (e.g., point cloud), they either rely on heuristics to find frontiers of the map, or predict views with AI models that optimize coverage. With the emerging implicit 3D reconstruction being able to reconstruct finer details of the complex 3D scene, optimizing accuracy is also an emerging requirement for NBV, where quantification of the information gain of the candidate views is needed.

Uncertainty Estimation

The information gain from training data for a training model can be directly modeled as the uncertainty reduction of model parameters, and such uncertainty estimation is a longstanding problem for machine learning. A classic framework for uncertainty estimation is the Bayesian Learning framework that estimates the posterior distribution of the model given the existing training data. However, such approaches typically require multiple model evaluations which are computationally expensive, and require significant modifications over network architectures and training procedures.

Recent work focus on the NeRF structure and approximate the uncertainty of model parameters with the posterior distribution of the output density and radiance (uncertainty of the model output). They typically follow the pattern of generalization of standard NeRF that learns a probability distribution over all the possible radiance fields modeling the scene, where an extra model head is online or offline trained to estimate the variance of the radiance fields under the supervision of existing views.

Intuitively, the standard deviation of distribution of model output stems from the model not attending to high-frequency details. However, the grided 3D reconstruction on the one hand minimizes such uncertainty by decomposition, on the other hand incurs an extra source of uncertainty caused by the capacity limit of local latent codes, which severely disturbs the NBV selection of these methods by incurring high uncertainty of model output and limited information gain. As a result, these methods tend to be stuck at certain spots where the corresponding local latent codes reach their capacity limit, causing inefficiency.

Using a finer-grained grid of latent codes seems to be a simple remedy, which avoids the local latent codes to reach their modeling capacity, but this remedy results in a much larger model size and GPU memory consumption and also, it is reported a too fine-grained grid of latent codes results in not enough training data input within each local region for training the local embedding, causing holes in the reconstructed model. In a word, the uncertainty caused by the capacity limit of local latent codes would be a common problem in grided implicit 3D reconstruction.

Methodology

Preliminary

Grided Implicit 3D Reconstruction Implicit 3D reconstruction can be defined as follows. A sequence of captured views $\boldsymbol{v_n} = \{v_0, v_1, ..., v_n\} \subseteq \boldsymbol{v}$, where $v_i = \{I_i, p_i\}$, I_i represents the image captured and p_i represents the corresponding camera pose. By assuming that training samples $v_i \in \boldsymbol{v_n}$ are independent of each other, we then optimize the whole latent codes $\theta = \{\theta_0, ..., \theta_m\}$ by minimizing a loss function $L(\theta, \boldsymbol{v_n}) = \frac{1}{n}\sum_{i=0}^{n} f(\theta, v_i)$. $f(\theta, v_i)$ is typically the negative log-likelihood between model output $r(\theta, p_i)$ and I_i, approximated by mean square error. Grided implicit 3D reconstruction follows similar definitions except that multiple light weightθ are grouped together as a grid and each maps a small region of the 3D scene. The grid of local latent codes are jointly optimized by summing up their loss functions.

NBV When exploring a previously unknown environment, NBV for implicit 3D reconstruction aims to to find a view $v_{n+1} \in \boldsymbol{v}$ that maximize information gain:

$$v_{n+1} = \arg\max_{v} g(\theta^n, v, \boldsymbol{v_n}) \tag{1}$$

where θ^n is the latent codes after optimization over $\boldsymbol{v_n}$. Note that NBV problem values sampling budget and it is commonly adopted that $I \in v$ is unknown when searching for NBV. Thus equation (1) is rewritten as

$$v_{n+1} = \arg\max_{v,p \in v} g(\theta^n, p, \boldsymbol{v_n}) \tag{2}$$

There might be various possible definitions for $g(\cdot)$ that calculates information gain of an unknown view v given θ_n and $\boldsymbol{v_n}$, but they should fundamental be in proportion to the possible improvement to θ towards an optimal θ' that minimizes $L(\theta, \boldsymbol{v})$, which is intuitively the knowledge about the whole scene extracted from v:

$$g(\theta^n, v, \boldsymbol{v_n}) \propto \sum_{i=0}^{m} (|\theta'_i - \theta^n_i| - |\theta'_i - \theta_i^{\boldsymbol{v_n}+v}|) \tag{3}$$

Prior work to find NBV for implicit 3D reconstruction model $g(\cdot)$ as the possible space for improvement for $r(\theta, p)$. Specifically, they model $r(\theta, p_i)$ as a fully parametric distribution whose parameters are learned by training the model with an extra model head and the uncertainty of $r(\theta, p_i)$ is estimated as the divergence inferred from the learned distribution. Such methods suffice to approximate changes to θ when the model capacity is enough, which means if there are views of non-zero uncertainty, their uncertainty can always be reduced after certain training iterations. However, such assumption does not hold in grided implicit 3D reconstruction, where the optimizable parameters are decomposed to a grid of small latent codes (e.g., 8 parameters) and each maps a comparatively simple local geometry. In this way, on one hand the local latent codes converge fast, causing limited space for uncertainty reduction, on the other hand local latent codes maybe unable to perfectly model certain local geometry due to limited capacity, resulting in comparatively high uncertainty with limited extracted knowledge, where such methods tend to be trapped. Also, to find NBV among candidate views, such methods need inference of the model for each candidate view, which is a non-negligible computation burden when the number of candidate views is large.

Training-Time Uncertainty Estimation

To save the computation burden of model inference for estimating information gain for each view, we consider the information that can be directly extracted from the latent codes. While it is difficult to compute $\theta_i^{\boldsymbol{v_n}+v}$ in equation (3) since I is unknown, inspired by parameter regularization methods introduced in continual learning, we can find how certain (or consolidated) θ_i^n is from the knowledge of $\boldsymbol{v_n}$ in previous training, which is inversely proportional to space for further changes of θ_i.

Since the training set is iteratively appended during the NBV process, we view training on $\boldsymbol{v_i}$ as a task T_i and consider the synaptic dynamics observed during this task. Similar to TODO, we assume the training and optimization of $L(\cdot)$ and θ during a task as a continual process on time t, and there exists an infinitesimal parameter update $\Delta\theta$ at time t which results in change in $L(\cdot)$ as

$$L(\theta + \Delta\theta, \boldsymbol{v_{i+1}}) - L(\theta, \boldsymbol{v_{i+1}}) \approx \sum_{j=0}^{m} \frac{\partial L}{\partial \theta_j} \Delta\theta_j \tag{4}$$

Thus, assume θ^i is computed at t_i and θ^{i+1} is computed at t_{i+1}, there is

$$\begin{aligned} L(\theta^{i+1}, \boldsymbol{v_{i+1}}) - L(\theta^i, \boldsymbol{v_{i+1}}) &\approx \sum_{j=0}^{m} \int_{t_i}^{t_{i+1}} \frac{\partial L}{\partial \theta_j(t)} \Delta\theta_j(t)\,\mathrm{d}t \\ &= \sum_{j=0}^{m} \omega_j^{i+1} \end{aligned} \tag{5}$$

The right part of equation (5) is equivalent to the path integral of the gradient along the parameter trajectory from θ^i to θ^{i+1}, which associates the records of gradients and parameter updates (synaptic dynamics) of each parameter with change in loss function and represents parameter specific contribution to changes in the total loss. In practice, ω_j^{i+1}

can be approximate using gradients and model updates computed during iterative mini-batch training. More about ω_j^{i+1} is discussed in TODO.

Since we are assuming efficient training on grided implicit 3D reconstruction, it is safe to assume $L(\theta^{i+1}, \boldsymbol{v_{i+1}}) - L(\theta^i, \boldsymbol{v_{i+1}})$ correctly extracts knowledge from $\boldsymbol{v_{i+1}}$, which means ω_j^{i+1} can be used as evidence about how much $\boldsymbol{v_{i+1}}$ consolidate θ_j^{i+1}. However, since ω_j^{i+1} is dependent to $|\theta_j^{i+1} - \theta_j^i|$, we cannot directly use it as a comparison metric across different parameters. Instead, we model the certainty of θ_j^{i+1} from $\boldsymbol{v_i}$ by as the multiplication of two dimensionless factors, influence factor h_1 and concentration factor h_2.

Let us consider the variation relationship between $L(\cdot)$ and θ_j by fixing other parameters in $L(\cdot)$ and we can get $L'(\theta_j, \boldsymbol{v_n})$. Since $\frac{L'(\theta_j^{i+1}, \boldsymbol{v_{n+1}}) - L'(\theta_j^i, \boldsymbol{v_{n+1}})}{\theta_j^{i+1} - \theta_j^i} = \tan\alpha$ represents the slope of $L'(\cdot)$ when θ_j varies from θ_j^i to θ_j^{i+1}, we have

$$h_1(\theta_j^{i+1}, \omega_j^{i+1}) = \frac{\omega_j^{i+1}}{\theta_j^{i+1} - \theta_j^i} \approx \frac{1}{\cos\alpha} \tag{6}$$

because ω_j^{i+1} represents the path trajectory. This is depicted in figure TODO. Intuitively, steeper slope (smaller $\cos\alpha$) means θ_j is more influential to the loss function and a larger value for $h_1(\cdot)$, making us more certain about θ_j^{i+1}.

For the concentration factor, we have

$$h_2(\theta_j^{i+1}) = \frac{1}{\theta_j^{i+1} - \theta_j^i} = \frac{sign(\theta_j^{i+1} - \theta_j^i)}{|\theta_j^{i+1} - \theta_j^i|} \tag{7}$$

where $\frac{1}{|\theta_j^{i+1} - \theta_j^i|}$ estimates how concentrated θ_j is and higher level of concentration (smaller $|\theta_j^{i+1} - \theta_j^i|$) intuitively represents higher certainty. $sign(\theta_j^{i+1} - \theta_j^i)$ estimates sign of $\theta_j^{i+1} - \theta_j^i$ to maintain the information about the optimization direction, in case that the optimization direction disagrees due to noises between certain tasks, such as in a task T_i, $\theta_j^{i+1} > \theta_j^i$ and in another task T_j, $\theta_j^{i+1} \leq \theta_j^i$.

Finally, we have the certainty estimation of θ_j^{i+1} in task T_{i+1}

$$c_j^{i+1} = \frac{\omega_j^{i+1}}{(\theta_j^{i+1} - \theta_j^i)^2 + \epsilon} \tag{8}$$

which is mostly a multiplication of $h_1(\cdot)$ and $h_2(\cdot)$, and a dampening factor ϵ is added in case that $\theta_j^{i+1} - \theta_j^i$ is close to zero. Summing up all the certainty estimation from training in the past, we have the certainty estimation of θ_j^{i+1} which is inversely proportional to information gain in equation (3)

$$C(\theta_j^{i+1}) = |\sum_{k=0}^{i+1} c_j^k| \propto |\theta'_j - \theta_j^{i+1}| - |\theta'_j - \theta_j^i| \tag{9}$$

Note that absolute value is used since different parameters can have different overall optimization direction. In this way, we can maintain an online certainty (or rather, uncertainty) estimation of each parameter in θ by aggregating the synaptic dynamics during training and ease the burden of any extra training or inference.

TUEV

To leverage the certainty estimation of each parameter in θ for NBV selection, we exploit the 3D correspondence relationship between local latent codes and the 3D scene that needs to be mapped. Intuitively, TUEV needs to find a view that visits local latent codes of the least certainty to maximize equation (3). However, for a view $v = \{I, p\}$, 3D correspondence of the local latent codes can be in three situations when rendered in I: empty without any geometries, having geometries rendered in I and occluded without any geometries rendered in I. Thus, we need to further carefully process $C(\theta_j^n)$ to get an accurate estimation of information gain in a candidate view. We define the set of local latent codes that are visible (not occluded) from any view in $\boldsymbol{v_n}$ as $\boldsymbol{\theta_{v_n}}$. This can be easily computed by transforming the coordinates of the local latent codes to the camera frames, projecting them to I using the intrinsic matrix of the camera and estimating whether their projections lie in I and depth testing.

To distinguish the above three situations of (the 3D correspondence of) the local latent codes, we create a sparse data grid G of the same 3D dimensions as the grid of local latent codes and assign three possible states (empty, surface and unknown) to the value of each data point on the grid. $\theta_j \notin \boldsymbol{\theta_{v_n}}$ are marked as unknown by initializing the corresponding point on G with a negative value. A negative value is naturally smaller than any $C(\theta_j^n)$ which is positive, and thus attract TUEV to explore unknown areas. $\theta_j \in \boldsymbol{\theta_{v_n}}$ are first initialized with 0 and are updated by $C(\theta_j^n)$ computed after training the latent codes over $\boldsymbol{\theta_{v_n}}$. Typically, the training process of grided implicit 3D reconstruction only trains latent codes that lies around surfaces in the 3D scene to save computation. As a result, θ_j lying in empty areas will be marked empty with a value of zero on G and θ_j around surfaces will be marked surface with a positive $C(\theta_j^n)$ value on G.

Given a candidate view $v = \{I, p\}$ where I is unknown and data grid G that stores the certainty estimation of the grid of local latent codes, we then render G to I' following the common rendering process of point cloud except treating unknown zones as surfaces $I' = Render(G, p)$, and average the non-zero values of I' as the mean certainty estimation, ignoring the empty zones. Summarizing the above process, our information gain estimation can be written as

$$g(\theta_n, v, \boldsymbol{v_n}) = -\frac{1}{K}\sum_{k=0}^{K-1} I''_k \tag{10}$$

where $I'' = \{I'_k | I'_k \in I' \wedge I'_k \neq 0\}$ and K is the size of I''. With equation (10), we can predict the information gain of v via more light-weight and efficient point cloud rendering process compared with model inference, making TUEV suitable for edge devices that are limited in computation resources. Equation (10) guides the selection of NBV in equation (1) to first cover the unknown areas and once unknown areas are all covered, revisit surfaces that are least certain, to get the greatest possible improvement of θ.

Experiments

Experimental Setup

Datasets We evaluated TUEV extensively on three types of datasets: complicated 3D models downloaded online[1], large 3D indoor scene from the Replica dataset and noised 3D indoor scene from ICL-NUIM. The downloaded 3D models are models of a drone, a motorcycle and a robot with their statistics listed in TODO and we generate circular poses of camera with a fixed radius and the camera pointing to the center of the 3D models. For the latter two type of datasets the camera can freely roam around in the indoor and simulated noise is added for the ICL-NUIM dataset. We generate a large set (sized 5000) of random camera poses and the corresponding rendered RGBD image from these poses as the whole set of possible view $\boldsymbol{v}$ for the former two datasets, and we use the default set of possible views in ICL-NUIM. All used datasets are provided with ground-truth 3D mesh models for evaluation.

Metrics We mainly follow the metrics used in TODO for end-to-end evaluation. Specifically, Accuracy (referred to as Accu.) measures the fraction of points from the reconstructed mesh that are within a preset distance from the ground-truth mesh. Completeness (referred to as Comp.) calculates the fraction of points from the groundtruth mesh that are within a preset distance from the point in the reconstructed mesh. F1 score (referred to as F1) is the harmonic mean of accuracy and completeness, which quantifies the overall reconstruction quality. Other metrics we used is Peak Signal-to-Noise Ratio (PSNR) $PSNR = 10 \cdot \log_{10}(\frac{MAX^2}{MSE})$, which measures the the similarity between the groundtruth image and the image rendered from implicit model. MAX is the maximal possible value on the image and MSE is the mean square error between the two images.

Workload We choose the state of the art 3D reconstruction method BNV Fusion as our major workload, which features a representative and efficient grided implicit 3D reconstruction pipeline. We use the default hyper-parameters in the demo setting in BNV Fusion and use their released checkpoint for the encoder and decoder. The only modification we made on BNV Fusion is to explicitly group their local latent codes as a matrix to simplify our implementation of TUEV.

Baselines

random, max distance, online model predicting output uncertainty, offline model predicting output uncertainty

End-to-End Results

time, number of views, accuracy, completeness, F1

Breakdown

number of views, PSNR reduction

Ablation Study

influence factor, concentration factor

Discussions

limitation, broader impact

Conclusion

References

[1] https://www.cgtrader.com