7ZF-02

リアルタイムレンダリング可能なNeRFの動的シーンへの拡張

武田 司 † 山口 周悟 † 岩瀬 翔平 † 佐藤 和仁 † 森島 繁生 ‡
† 早稲田大学 ‡ 早稲田大学理工学術院総合研究所

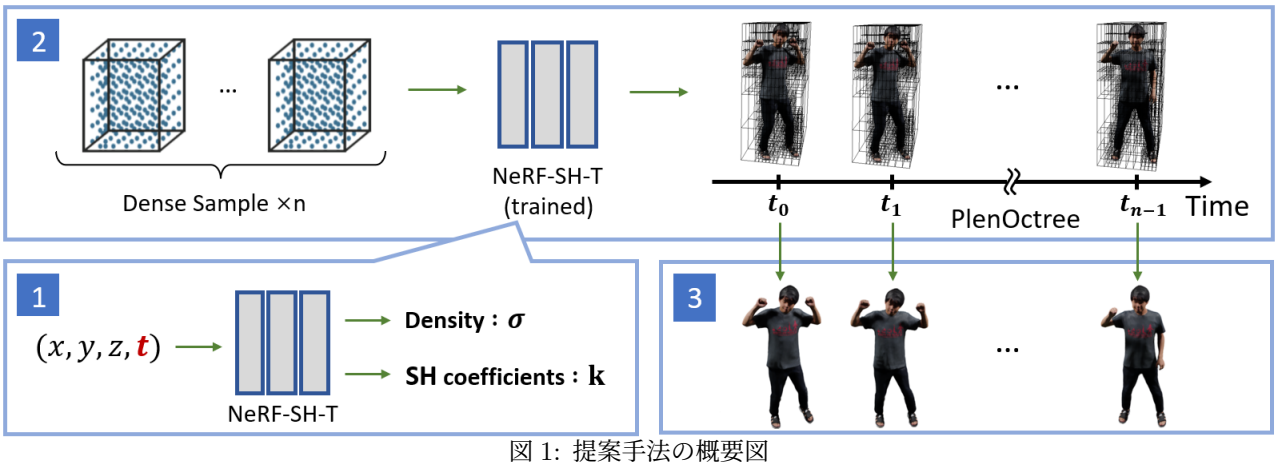

図 1: 提案手法の概要図

1. はじめに

2021 年、東京で開催されたオリンピックのスポーツ中継で自由視点映像が用いられるなど、撮影を行ったカメラの視点とは異なる新たな視点からの画像生成を行う、Novel View Synthesis (NVS) の需要が高まってきている。その中で、3D 正解データを必要とせず、カメラポーズ付き多視点画像のみで NVS を行う手法として、Neural Radiance Fields(NeRF) [3] が注目を集めている。NeRF は、空間上の各点に輝度値・密度を対応づけ、古典的なボリュームレンダリング [1] の手法を用いることで、各視点に対応した画像を生成する。この手法では、生成された画像が実際の画像と一致するようにニューラルネットワークの重みを学習することで、高精細な NVS を行うことが可能になる。しかし、NeRF は基本的に対象が静的なシーンに限定されることや、計算コストが高く、レンダリング時間が長い等の制約がある。

そこで本稿では、静的なシーンに限定されるものの、NeRF のレンダリング時間を大幅に高速化した Yu らの手法 [5] を動的シーンに拡張することで、上記 2 つの制約を解消することを試みる。そこで研究の流れとして、まず (1) 輝度値ではなく球面調和係数を出力する、NeRF-SH の時刻を加えた学習を行い、(2) 八分木表現である PlenOctree を時刻分生成する。加えて、(3) レンダラーを時刻方向に拡張することで、動的シーンにおける NeRF のレンダリング時間の高速化を目指す。

2. 関連研究

2.1 Neural Radiance Fields

NeRF [3] は、$F_\Theta : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$ と表されるように、3 次元空間上の位置 $\mathbf{x} = (x, y, z)$ と視線方向 $\mathbf{d} = (\theta, \phi)$ を入力として、輝度値 $\mathbf{c} = (r, g, b)$ と密度 σ を出力するニューラルネットワーク (以下 NN と呼ぶ) である。この NN の出力から、輝度値・密度の積を光線上で積分するボリュームレンダリング [1] を行うことで、任意の視点に対応した画像をレンダリングすることが出来る。NeRF は、レンダリングした画像と、実際の画像の差分を損失関数として、NN の重みを学習する。

この手法では、高品質な NVS が可能な反面、基本的に対象を静的なシーンに限定していることや、レンダリング時間が長い等の制約がある。後者の理由としては、レンダリングの際にも NN を通して輝度値・密度を求める必要があり、解像度が 800 × 800 の画像を生成する場合は計 1 億回以上 NN にクエリする必要があることが挙げられる。そこで、これらの制約を解消した派生手法も活発に研究されている。

D-NeRF [4] や、DyNeRF [2] は、NeRF のネットワークに時刻情報を組み込むことで、動的シーンへの拡張を実現した。また Yu らの手法 [5] はシーン情報を予め保存しておくことで、リアルタイムレンダリングを可能にした。次章ではこの手法について述べる。

2.2 PlenOctrees

Yu らは、オリジナルの NeRF [3] とは異なり、位置 $\mathbf{x}$ のみの入力から、輝度値を表現する球面調和関数の係数および密度を出力する NeRF-SH(式 (1)) を提案した。

$$f : (\mathbf{x}) \longrightarrow (\mathbf{k}, \sigma) \ \ where \ \ \mathbf{k} = (k_l^m)_{l:0 \le l \le l_{max}}^{m:-l \le m \le l} \qquad (1)$$

NeRF-SH は式 (2) に示すように、得られた球面調和関数に視線方向 $\mathbf{d}$ を与えることで輝度値が求まり、通常の NeRF 同様に学習を行うことが可能である。

$$c(\mathbf{d}; \mathbf{k}) = S\Big(\sum_{l=0}^{l_{max}} \sum_{m=-l}^{l} k_l^m Y_l^m(\mathbf{d}) \Big) \qquad (2)$$

ここで S はシグモイド関数、l はモデルの次数、m は位数を表している。この学習済みの NeRF-SH により得られた密度分布を元に空間を八分木構造に分割し、各グリッドに NeRF-SH により求まる球面調和係数および密度を保存する。この八分木のデータ構造を以降 PlenOctree と呼ぶ。加えて、PlenOctree を直接 Fine-tuning することで、学習時間が短くとも高品質な NVS が可能となる。ここで NeRF-SH を用いて球面調和係数を保存する

Extend NeRF for real-time rendering to dynamic scenes:
Tsukasa Takeda†, Shugo Yamaguchi†, Shohei Iwase†, Kazuhito Sato†, and Shigeo Morishima‡ (†Waseda University, ‡Waseda Research Institute for Science and Engineering)

理由としては、球面調和係数は視線方向に依存しないため、視点依存の効果を保持しつつ、コンパクトな形でシーン情報を保存するためであると考えられる。以上より、Yu らの手法は NN を通すことなく、輝度値・密度を得ることができるため、オリジナルの NeRF と比較して大幅な高速化を実現した手法である。しかし、この手法は NeRF 同様時間情報を扱っていないため、基本的に対象が静的シーンに限定されるという制約がある。

3. 提案手法

本稿では、対象が静的シーンに限定された Yu らの手法 [5] において、PlenOctree を時刻分生成することで、動的シーンへ拡張することを提案する (図 1)。まず PlenOctree の時刻分生成にあたって、各時刻において独立に NeRF-SH(式 (1)) を学習し、PlenOctree を生成することも可能である。しかし、この方法では NeRF-SH の学習時間が単純に時刻数倍となってしまい、膨大な時間を要してしまう。そこで我々は、(1)NeRF-SH の入力に新たに時刻を加え、全時刻をまとめて 1 つの時間変化する場として表現することで、学習時間の短縮化を試みる。ここで NeRF-SH の入力に離散的な時刻 t を加える新たな NeRF を NeRF-SH-T と呼び、このネットワークを式 (3) で与える。

$$f:(\mathbf{x},t)\longrightarrow(\mathbf{k},\sigma)\ \ where\ \ \mathbf{k}=(k_l^m)_{l:0\leq l\leq l_{max}}^{m:-l\leq m\leq l} \quad (3)$$

NeRF-SH-T の学習後、(2) 時刻単位で PlenOctree を生成し、Fine-tuning を行う。最終的に、(3) 各時刻の PlenOctree を順々に読み込み、レンダリングを行う。本研究では、各時刻において多視点画像が存在するという実験設定のもと、レンダリング速度の検証を行う。

4. 実験

4.1 実験概要

提案手法におけるレンダリング速度の高速性を確かめるため、本稿では NeRF-SH-T との比較実験を行った。この理由として、既存の動的シーンに対応した NeRF [2], [4] はレンダリング速度の高速化に焦点を当てておらず、NeRF [3] と同等、またはそれ以上の時間を要する。加えて、Yu ら手法 [5] によると、NeRF, NeRF-SH の性能はほぼ同レベルで、NeRF-SH-T のレンダリング時間はベースとなる NeRF-SH と同等になるためである。そこで、各時刻で多視点画像が存在するという設定のもと実験を行うため、Blender により各時刻において、解像度が 800 × 800 の画像 50 枚、計 30 時刻分のデータセットを作成し、学習に用いた。加えて、PlenOctree の解像度は $512\times512\times512$、球面調和関数における次数を $l=3$ とし、両手法とも、学習とレンダリングにおける GPU は Nvidia RTX 2080Ti を用いて行った。

4.2 結果

動的シーンのレンダリングを行った結果を図 2 に示した。加えてレンダリング速度 (FPS)、客観評価指標 (PSNR, SSIM, LPIPS) の計測を行い、表 1 に値を示した。ただし、提案手法におけるレンダリング速度は PlenOctree の読み込み時間も含めた値である。

表 1: 実験結果の比較

手法	FPS↑	PSNR↑	SSIM↑	LPIPS↓
NeRF-SH-T	0.036	36.10	0.988	0.036
提案手法	**1.607**	**36.96**	**0.991**	**0.029**

図 2: 動的シーンのレンダリング結果

この結果から、本研究では NeRF-SH-T と比較して、PlenOctree の Fine-tuning により品質を上回りつつ、レンダリング速度を約 50 倍高速化することを確認した。なお、提案手法では各時刻に対して PlenOctree を読み込む必要があるため、1 つの PlenOctree の読み込みで任意の視点のレンダリングを行える Yu らの手法 [5] と比較して、より多くの時間をレンダリングに要してしまうという結果になった。

5. おわりに

本稿では、カメラポーズ付き多視点画像のみで高品質な Novel View Synthesis を実現した NeRF において、リアルタイムレンダリングを実現した Yu らの手法 [5] を動的シーンへ拡張した際のレンダリング速度を検証した。実験により、提案手法はオリジナルの NeRF を動的シーンに適応する場合と比較して、レンダリング時間を約 50 倍高速化可能であることが分かった。しかし PlenOctree の容量が大きく、読み込みに多くの時間を割いてしまうため、リアルタイム性は保持できないという結果になった。そこで、今後は PlenOctree を時刻方向にも分割を加え、全時刻を 1 つの PlenOctree でまとめて表現するなど、よりコンパクトな形でシーン情報を保存する手法を開発することが、動的シーンのリアルタイムレンダリングを実現する方向性として考えられる。

謝辞

この研究は、JST 未来社会創造事業 (JPMJMI19B2) および JSPS 科研費 (19H01129,19H04137, 21H05054) の補助を受けています。

参考文献

[1] Drebin Robert A. *et al.* “Volume rendering”. *ACM SIGGRAPH Computer Graphics*, pp. 65–74, 1988.

[2] Li T. *et al.* “Neural 3D Video Synthesis”. *arXiv preprint arXiv:2103.02597*, 2021.

[3] Mildenhall B. *et al.* ”NeRF: Representing Scenes as Neural radiance Fields for View Synthesis”. In *European Conference on Computer Vision*, 2020.

[4] Pumarola A. *et al.* “D-NeRF: Neural Radiance Fields for Dynamic Scenes”. In *Conference on Computer Vision and Pattern Recognition*, 2021.

[5] Yu A. *et al.* “PlenOctrees for Real-time Rendering of Neural Radiance Fields”. In *International Conference on Computer Vision*, 2021.